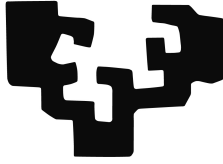


eman ta zabal zazu



UNIVERSITY OF THE BASQUE COUNTRY  
Computer Languages and Systems

PhD Thesis

---

**Computational Models  
for  
Semantic Textual Similarity**

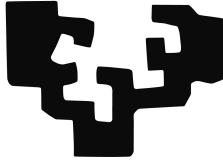
---

Aitor Gonzalez-Agirre

2017



eman ta zabal zazu



UNIVERSITY OF THE BASQUE COUNTRY  
Computer Languages and Systems

# **Computational Models for Semantic Textual Similarity**

PhD dissertation written by Aitor  
Gonzalez-Agirre, with the supervision of  
German Rigau i Claramunt and Eneko Agirre  
Bengoa.

Donostia, July 2017.



*Lo más importante para avanzar en la ciencia es la curiosidad. No pretendo aseverar que el conocimiento no es importante, pero un conocimiento demasiado arraigado puede provocar el efecto contrario, sobretodo si no es correcto, o al menos no lo es del todo. Cuestionar el conocimiento ha sido fuente de inagotables descubrimientos, de modo que es esencial que lo sigamos haciendo. Es posible aprender por casualidad, por imitación, o por otras causas, pero sin curiosidad no puede haber avances significativos.*

*Deseo dedicar este trabajo a un pequeño curioso, mi hijo Aimar, con la esperanza de que algún día pueda servirle de inspiración, y que ame la ciencia tanto como yo la amo. Aprende de todo. Cuestiónalo todo. Y sobretodo, se curioso.*

**Zure aita**



# Acknowledgements

First, I would like to thank my thesis advisors German Rigau and Eneko Agirre, because without their support and guidance I would not have been able to finish this research.

I would like to thank Dr. Mark Stevenson and his group in the University of Sheffield for their hospitality during my stay there in 2015.

I also want to thank to the IXA NLP group and my colleagues in the group for supporting and helping me.

Finally, I must express my sincerest gratitude to my parents and specially to my girlfriend for their continuous support throughout my many years of study and research. This thesis would not have been possible without them. Thank you!

## **Institutional acknowledgements**

This research is supported by a doctoral grant from MINECO (FPU12/06243). It has also been funded by the European Commission projects MEANING (IST-2001-34460), KYOTO (ICT-2007-211423), PATHS (FP7-ICT-2009-6-270082), OpeNER (ICT-2011-296451) and NewsReader (ICT-2011-316404) and the Spanish Government Projects KNOW (TIN2006-15049-C03-03), KNOW-2 (TIN2009-14715-C04-04), READERS (PCIN-2013-003-C02-02) and TUNER (TIN2015-65308-5-1-R).





# Abstract

Measuring semantic similarity between textual items (words, sentences, paragraphs or even documents) is a very important research area in Natural Language Processing (NLP). It has many practical applications in other NLP tasks such as Word Sense Disambiguation, Textual Entailment, Paraphrase detection, Machine Translation, Summarization, Information Retrieval or Question Answering.

The overarching goal of this thesis is to advance on computational models of meaning and their evaluation. To achieve this goal we define two tasks and develop state-of-the-art systems that tackle both tasks: Semantic Textual Similarity (STS) and Typed Similarity.

STS aims to measure the degree of semantic equivalence between two sentences by assigning graded similarity values. This graded similarity captures the notion of intermediate shades of similarity ranging from pairs of text that differ only in minor nuanced aspects of meaning, in relatively important differences, down to pairs that share only some details or that only have in common being about the same topic. In the scope of this research, we have collected pairs of sentences to construct datasets for STS, a total of 15,436 pairs of sentences, being by far the largest collection of data for STS.

Using these new datasets for STS we have designed, constructed and evaluated a new approach to combine knowledge-based and corpus-based methods using a cube. This new system for STS is on par with state-of-the-art approaches that make use of Machine Learning (ML) without using any of it, but ML can be used on this system, improving the results.

Typed Similarity tries to identify the type of relation that holds between a pair

---

of similar items in a digital library. Being able to provide a reason why items are similar has applications in recommendation, personalization, and search. We investigate the problem within the context of Europeana, a large digital library containing items related to cultural heritage. A range of types of similarity in this collection were identified and a set of 1,500 pairs of items from the collection were annotated using crowdsourcing.

Finally, we present three systems capable of resolving the Typed Similarity task: a baseline approach, a knowledge-based approach and a ML system. The high results obtained by our systems suggests that this technology is close to practical applications. In fact, the system based on ML resulted in a real-world application to recommend similar items to users in an online digital library.

# Contents

<b>Abstract</b>	<b>vii</b>
<b>Contents</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research framework . . . . .	1
1.2 Main goals . . . . .	4
1.3 Main contributions . . . . .	5
1.4 Document structure . . . . .	7
<b>2 Background</b>	<b>9</b>
2.1 Semantics . . . . .	9
2.2 Word Similarity . . . . .	12
2.3 Knowledge-based word similarity . . . . .	13
2.3.1 Knowledge-bases and resources . . . . .	14
2.3.2 WordNet-based methods . . . . .	15
2.3.3 Wikipedia-based methods . . . . .	18
2.4 Corpus-based word similarity . . . . .	19
2.4.1 Distributional Semantics . . . . .	19
2.4.2 Distributed Representations and Neural Networks . . . . .	22
2.4.3 Continuous Bag-of-Words and Skip-gram models . . . . .	23
2.5 Combining Knowledge-based and Corpus-based similarity . . . . .	25
2.6 From Word Similarity to Textual Similarity . . . . .	26

## CONTENTS

---

2.7	Evaluation . . . . .	28
2.8	Best systems for Semantic Textual Similarity . . . . .	30
2.8.1	DKPro . . . . .	31
2.8.2	Takelab . . . . .	32
2.8.3	Ebiquity-Core . . . . .	32
2.8.4	DLS@CU . . . . .	33
2.8.5	Samsung-Ensemble . . . . .	34
2.9	Conclusions . . . . .	35
<b>3</b>	<b>Semantic Textual Similarity</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Design of the STS task . . . . .	38
3.2.1	Related datasets and tasks . . . . .	39
3.3	Gathering datasets for STS . . . . .	40
3.3.1	STS 2012 datasets . . . . .	41
3.3.2	STS 2013 datasets . . . . .	44
3.3.3	STS 2014 datasets . . . . .	46
3.3.4	STS 2015 datasets . . . . .	48
3.3.5	STS 2016 datasets . . . . .	49
3.4	Annotation . . . . .	52
3.4.1	Quality of annotation . . . . .	53
3.5	System Evaluation . . . . .	56
3.5.1	Evaluation metrics . . . . .	56
3.5.2	The baseline system . . . . .	59
3.5.3	Participation . . . . .	60
3.5.4	Results . . . . .	61
3.5.5	Tools and Resources . . . . .	63
3.6	Conclusions . . . . .	63
<b>4</b>	<b>Cubes for Semantic Textual Similarity</b>	<b>67</b>
4.1	Motivation . . . . .	67
4.2	Building Cubes . . . . .	68
4.2.1	Layers of the cube . . . . .	73
4.3	Producing the STS Score . . . . .	78
4.3.1	Pairwise similarity score . . . . .	78
4.3.2	Threshold . . . . .	80
4.3.3	Hierarchical cube . . . . .	83
4.3.4	Machine Learning . . . . .	84
4.4	Evaluation . . . . .	86

---

4.4.1	Train	87
4.4.2	Development	89
4.4.3	Test	91
4.4.4	Ablation test	93
4.5	Conclusion	94
<b>5</b>	<b>Typed Similarity</b>	<b>97</b>
5.1	Introduction	97
5.2	Europeana	100
5.3	A dataset for typed similarity	102
5.3.1	Defining similarity types	102
5.3.2	Selecting item pairs	103
5.3.3	Annotation	103
5.3.4	Quality of annotation	104
5.4	Discussion and analysis	109
5.5	Systems evaluation	113
5.5.1	Evaluation metrics	114
5.5.2	The baseline system	115
5.5.3	Results	115
5.6	Conclusions	117
<b>6</b>	<b>A System for Typed Similarity</b>	<b>119</b>
6.1	Introduction	119
6.2	Similarity methods	120
6.2.1	TF-IDF	121
6.2.2	LDA	121
6.2.3	WLVM	122
6.2.4	Random walks	123
6.3	Constructing systems	124
6.3.1	Processing text in the items	124
6.3.2	Baseline system	124
6.3.3	Knowledge based approach	125
6.3.4	Machine learning system	127
6.4	Evaluation	127
6.4.1	Train and Development	127
6.4.2	Test	129
6.4.3	Comparison to the best system	132
6.5	Conclusion and Future Work	132

## CONTENTS

---

<b>7</b>	<b>Conclusions and future work</b>	<b>135</b>
7.1	Summary . . . . .	135
7.2	Publications . . . . .	138
7.3	Future work . . . . .	141
	<b>Bibliography</b>	<b>143</b>

# Introduction

This introductory chapter is organized in four sections. First, Section 1.1 introduces the research framework and presents some examples to introduce the reader in the notion of semantic similarity. After that, Section 1.2 describes the main goals of this research, and Section 1.3 presents the main contributions of the research. Finally, in Section 1.4 we describe the structure of the rest of the document.

## 1.1 Research framework

Communicating verbally with machines has been one of the main objectives since the birth of computing. In 1968 Arthur C. Clarke made half the planet's imagination run free with his novel '*2001: A Space Odyssey*', which was transferred concurrently to the big screen by Stanley Kubrik with great success. Computer science had barely taken its first steps (the first microprocessor had not been developed yet), but the idea of an artificial intelligence like *HAL 9000* had already seduced a generation that had not even touched a personal computer.

As usual in scenarios like this, well into 2017, we are far from replicating the communicative skills that *HAL 9000* was supposed to have in 2001. Understanding human language may seem very simple, people do it every day, but it is a very difficult task for machines. The language is full of phenomena that make comprehension very complex: polysemy, irony, sarcasm, double meanings, multiple ways of saying the same thing... And if this were not enough, understanding depends on our knowledge of the world, which we use to reason and understand each other, in a process that we do practically without realizing it. A clear example could be

## CHAPTER 1. INTRODUCTION

---



(a) Most probable interpretation for the sentence 'Fred saw a plane flying over New York'.



(b) Very unlikely interpretation for the sentence 'Fred saw a train flying over New York'.



2 (c) Most probable interpretation for the sentence 'Fred saw a train flying over New York': a plane flying over New York and someone seeing a train on the ground.

**Figure 1.1** – Example of possible interpretations for two almost identical sentences.



the following two sentences:

- *Fred saw a plane flying over New York.*
- *Fred saw a train flying over New York.*

Although both sentences only change in one word, they have very different *meanings*. It is easy for us to imagine a plane flying over New York (Figure 1.1a), but as we read the second sentence, we soon realize that a train does not fly (Figure 1.1b), and therefore it means something different, such as a plane flying over New York with a person inside it (because people do not fly either), who is seeing a train from the plane window (Figure 1.1c). Making these kinds of inferences is very complicated for computers, but relatively simple for people. Another example is the following:

- *Fred saw the plane flying over Berna.*
- *Fred watched the jet soaring over the capital of Switzerland.*

The previous two sentences are very similar even though they are realized very differently. It only seems that the second sentence is more precise than the first one. But both *meanings* are compatible. A computer system capable of recognizing that 'saw' and 'watched', 'plane' and 'jet', 'flying' and 'soaring' are very similar and that 'Berna' is the 'capital of Switzerland' could evaluate correctly that these two sentences are equivalent.

These measures of equivalence of meaning are useful for many tools, for example the well-known *Siri*, iOS' personal assistant, or to evaluate voice commands in a home automation system, so that the house can deduce that the sentence '*I need more light*' can mean '*Raise the blinds*' or '*Turn on the lights*', depending on the light conditions in the outside at that moment. Other possible applications are helping the elderly, since in general people in this age range have greater difficulties in learning and using complicated user devices and may prefer voice commands, or in the *teaching domain*, where a STS system is capable to evaluate whether the student's response mean the same as the correct answer assigned by the teacher, facilitating the task to the teacher. STS can also help in *Machine Translation*, increasing the variability of the translations, generating sentences written differently but without changing the original meaning. In a similar way, STS can help in other tasks such as *Plagiarism detection* or *Question Answering*.

## 1.2 Main goals

In linguistics, *semantics* is the study of the *meaning* of words, their structure, and their relationships with other words. *Meaning* is the mental representation of an object or concept, what we see in our mind when we see or hear it. *Evaluating the meaning* is an important part of *Natural Language Processing* (NLP), a field of computer science, artificial intelligence and computational linguistics with the objective of understanding and generating human language. *Natural Language Understanding* (NLU) consist of a program reading a text and constructing from it a conceptual representation of its meaning. NLU requires multiple processes including morphological, syntactic, semantic and pragmatic analysis of languages. The overarching goal of this thesis is to advance on computational models of meaning and their evaluation.

*Semantic Textual Similarity* (STS) is a task originally presented in 2012 at the *International Workshop on Semantic Evaluation* (SemEval 2012), which addresses one of the aspects of NLP and artificial intelligence that will allow machines to communicate naturally with people: the *comparison of meaning*. Given two sentences, knowing if both have the same meaning or not is crucial for good communication. But the meaning is not white or black, it has a whole variety of gray tones. STS aims to automatically evaluate the similarity between sentences in a scale from 0 to 5, and in which each range represents in an easily understandable way the differences that make those two sentences equivalent or not. The first goal of this thesis is to design the STS task and to create and annotate datasets for it.

Another goal of this thesis is to investigate how to create systems capable to solve this task. There are many techniques and resources that are useful for that purpose. Among resources, WordNet, Wikipedia and ontologies such as SUMO allow to address many of the difficulties mentioned above such as polysemy, detection of entities (people, companies, etc.) or reasoning. Recently, thanks to the rise of *Deep Learning*, new useful resources (such as *word embeddings*) have been automatically generated, improving the performance of STS systems. These word embeddings are capable of storing the semantic characteristics of words in a vector, where the vectors of words with similar meanings are closer and different ones further away. These new systems analyse the context in which words appear in a very large corpora, and assuming that similar words appear in similar contexts, places each word in an N-dimensional space, keeping the premise that similar words should remain close one to another. Once these vectors are generated the similarity between words can be computed by calculating the cosine of

their vectors.

In addition to comparing the meaning of two sentences, humans are able to reason why or in what sense those sentences (e.g. a description of an object) are similar. In other words, similarity can be measured in a generic way, as we have seen so far, but can also be decomposed into different types. Measuring the type of similarity is very useful for recommendation systems. With this goal, this thesis also presents another *SemEval* task, *Typed Similarity*, that aims to elucidate the type of similarity between items on an online portal of cultural heritage items. In this task eight different types of similarity are defined, like similar *author*, similar *location* or similar *time period*.

Within the framework of this task, we also present a system capable of providing similar items to users based on the similarity type. This way, users can follow a proposed path, visiting the items of the online museum according to his preferences (for example visiting objects of similar periods or similar civilizations). This is useful for day-to-day applications, like the recommendation systems of many online shopping platforms such as Amazon or Ebay. For example, if you visit a book on these platforms the recommender system can offer you books with a similar topic, that takes place at a similar period or written by similar authors, not just books by the same author or the same publisher. The last goal of this research is to design a system capable of identifying similarity types.

In summary, throughout this thesis STS and Typed Similarity tasks are designed, their principles are defined, and the necessary datasets are created. Following to the definition of the tasks, we present systems capable of solving these tasks with a high performance.

## 1.3 Main contributions

When the work presented in this thesis began, STS was not defined yet. In this period it has become a recognized task with great acceptance. In addition to participating in the design and organization of the STS task, my contributions include the design and organization of *Typed Similarity*. Throughout this time, various aspects of the task have been polished, and we have been able to reach consensus with the scientific community on the task.

This research have demonstrated that the task is feasible, and several systems capable of evaluating sentences have been created within the framework established by STS. I have also provided one system for STS and another for Typed Similarity. Both systems achieve state-of-the art results, and the system for typed similarity has proved to have practical applications in the real world.

Summarized, the main contributions of this thesis are:

- **Definition of STS:** My contribution involves the definition of STS as a task where given two snippets of text, system assign a graded similarity score ranging from 0 to 5. STS has achieved a great acceptance, becoming the most popular task at SemEval, also being used to evaluate *sentence representations* or *sentence embeddings*.
- **Definition of Typed Similarity**, a new task related to STS. Typed Similarity defines eight similarity types and tries to determine these types of similarity between cultural heritage items in an online digital library.
- **Datasets creation:** We have collected pairs of sentences to construct **datasets for STS**, which after five years make a total of 15,436 pairs of sentences, being by far the *largest collection* of data for STS. These datasets are manually annotated with high quality using Crowdsourcing. This involved the design and implementation of gold-standard pairs to control the annotations from turkers, and mechanisms to filter them detecting outliers (or bad annotators) and improve inter-tagger correlations. We automatically gathered cultural heritage items pairs to construct the **datasets for Typed Similarity**, and annotated them in the same way as STS dataset. All these datasets are freely available in the STS Wiki<sup>1</sup>.
- We have designed, constructed and evaluated a **new approach to combine knowledge-based and corpus-based methods** using a cube. This new system is on par with ML approaches without using any of it, and using ML on this system improves the results further. As part of this work we have analysed the most used resources, methods, and algorithms to perform STS. We did intensive experiments to evaluate the quality and usefulness of these resources. Additionally, we have carried out a comparison between our system based on the cube and the typical STS systems, and an analysis of the main differences.
- We have designed, constructed and evaluated a **system capable of resolving the Typed Similarity** task. This system resulted in a real-world application to recommend similar items to users in an online digital library. Error analysis is carried out for the Typed Similarity system, with the objective of identifying the main issues that can help in the design of better systems for the task in the future.

---

<sup>1</sup>[http://ixa2.si.ehu.es/stswiki/index.php/Main\\_Page](http://ixa2.si.ehu.es/stswiki/index.php/Main_Page)

- The **organization** of both tasks, which includes submitting the proposal to *SemEval* organizers, announcing it to participants, preparing the Train and Test sets for each of the year (including instructions), evaluating participant systems, performing an analysis of system and results, and writing the final task paper.

## 1.4 Document structure

This dissertation is organized in the following chapters:

- **Chapter 2: Background**  
This chapter provides an in depth review of different methods and resources to compute the semantic similarity between textual items. Furthermore, it presents several methods proposed for computing semantic similarity between words or texts, and introduces the dataset for semantic similarity available prior to the beginning of this thesis. Finally, it overviews the current best systems for semantic similarity.
- **Chapter 3: Semantic Textual Similarity**  
This chapter describes the Semantic Textual Similarity (STS) task.
- **Chapter 4: Cubes for Semantic Textual Similarity**  
This chapter presents a novel system for STS that can combine several resources, forming a cube where each resource is added as a layers, and its comparison with the state-of-the-art.
- **Chapter 5: Typed Similarity**  
This chapter presents the Typed Similarity (Typed STS) task, that aims to identify the type of relation that holds between a pair of similar items in a digital library.
- **Chapter 6: A System for Typed Similarity**  
This chapter describes a system for identifying Typed Similarity, and how it is used in Europeana to recommend similar items to the users.
- **Chapter 7: Conclusion and Future Work**  
This chapter draws the main concluding remarks and provide some lines for future research.



## Background

This chapter provides a revision of the state-of-the-art on computational lexical semantics for Natural Language Processing (NLP) and presents several methods proposed for computing semantic similarity between words or texts. First we briefly introduce semantics and semantic similarity in Sections 2.1 and 2.2. Next, we describe Knowledge-based and Corpus-based similarity in Sections 2.3 and 2.4, and how to combine them in Section 2.5. In Section 2.6 we explain how to extend word similarity to textual similarity. Next, Section 2.7 introduces the datasets for semantic similarity available prior to the beginning of this thesis. In Section 2.8 we review the current best systems for semantic similarity. Finally, Section 2.9 draws some conclusions.

### 2.1 Semantics

Linguistics is the study of the sounds, grammar and meaning in languages. The final objective in linguistics is to explain why patterns in languages are as they are. Linguistics tries to explain why phenomena occur in languages using a *descriptive approach*, and find the rules people unconsciously follow when they speak and write. On the other hand, *prescriptive approaches* try to describe how people should speak and write and what rules of language people should know. In linguistics, **semantics** is the study of the meaning of words, their structure, and their relationships with other words.

*Meaning* is the mental representation of an object or concept, what we see in our mind when we see or hear it. The complete meaning of a word is always contextual, and no study of meaning separated from context can be taken seriously

## CHAPTER 2. BACKGROUND

---

(Firth 1935). In general we can distinguish two types of meaning, *lexical meaning* and *grammatical meaning*. The former is the meaning of all words that contain a lexeme: nouns, verbs, adjectives and some adverbs. Grammatical meaning is the meaning of a word in relation to its function in the sentence, such as articles, determiners, prepositions or pronouns.

*Lexical Semantics* is a subfield of semantics that studies the meaning of individual words and their relationships. In other words, the study of *Lexical Units* (also called syntactic atoms). Lexical Units are the basic elements of a lexicon, the vocabulary of a language, and they constitute the minimal meaning units.

Some words can have several different meanings, also known as *senses*. This phenomenon is called *polysemy*. An example of polysemy is the word bank, which can mean 'a huge bank of earth' or a 'financial institution'. Polysemy is the opposite of *monosemy*, only having one meaning per word. The sense of a word is a widely accepted meaning for that word. For example, these are the senses or meanings for the word bank in WordNet 3.0 (Fellbaum 1998):

1. sloping land (especially the slope beside a body of water): *they pulled the canoe up on the bank.*
2. a financial institution that accepts deposits and channels the money into lending activities: *he cashed a check at the bank.*
3. a long ridge or pile: *a huge bank of earth.*
4. an arrangement of similar objects in a row or in tiers: *he operated a bank of switches.*
5. a supply or stock held in reserve for future use (especially in emergencies)
6. the funds held by a gambling house or the dealer in some gambling games: *he tried to break the bank at Monte Carlo.*
7. a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force
8. a container (usually with a slot in the top) for keeping money at home: *the coin bank was empty.*
9. a building in which the business of banking transacted: *the bank is on the corner of Nassau and Witherspoon.*



10. a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning): *the plane went into a steep bank.*

Another example of a *polysemic* word is **wood**:

- *a piece of a tree.*
- *a geographical area with many trees.*

When humans read sentences involving the words **bank** or **wood** (and other words) they find it easy to infer their meaning through its context, by using their knowledge of the world. Most words are polysemic, and the more polysemic a word is, the more frequently it is used (Zipf 1932).

*Homonymy* is a phenomenon that is often confused with polysemy. Homonymous words are those that are pronounced in the same way, but whose meaning is different. These words can be spelled the same or not, such as 'bark' (the sound of a dog) and 'bark' (the skin of a tree), or 'too' and 'two'. Consider as another example the following sentences:

- *I traveled by train from Barcelona to Donostia.*
- *Mikel and Joseba train every day at the gym.*

The word **train** has different meaning in the above sentences:

- *a series of connected railway carriages or wagons moved by a locomotive or by integral motors.*
- *to make (a person) fit by proper exercise, diet, practice, etc., as for an athletic performance.*

Therefore, we can say that there are words that are spelled in the same way but with different senses. Basically, a sense is one of the possible meanings of a given word. If two different senses of a word are *not semantically related* between them we are talking about a homonymy relation, as the example with train we just saw. Instead, if two senses of a word are *semantically related* we are talking about polysemy.

When the meaning of a sentence does not change when substituting a word for a different one they are said to be *synonymous*. It is not clear if true synonyms exist, since although in most contexts one word can substitute another, there may be some context where they are not interchangeable.

*Antonymy* is the opposite of synonymy. Two words are antonyms if the meaning of one is the opposite of the other, as *'expensive'* and *'cheap'*.

Meaning has a hierarchical structure, and the meaning of some words is included in other words, in a phenomenon called *hypernymy*. When the meaning of a word includes the meaning of another word we say that the first is the hypernym of the second. For example, *'animal'* is the hypernym of *'cat'*. Therefore, a hypernym is a more general and applicable term (less concrete).

*Hyponymy* is the inverse phenomenon of hypernymy. However, a hyponym may be the hypernym of other words. For example, *'mammal'* is hyponym of *'animal'*, but hypernym of *'cat'*.

*Meronymy* occurs when a word is part of another word. For example, *'finger'* is a meronym of *'hand'*, and *'tire'* is a meronym of *'car'*.

Finally, we can say that concepts that *share some meaning* are **semantically similar**. For example, *'dog'* and *'cat'* are more semantically similar than *'house'* and *'train'*. But if we compare *'dog'* and *'cat'* with *'car'* and *'bus'* the thing is not so clear: *'cat'* and *'dog'* are pets, and both *'car'* and *'bus'* are on wheels means of transportation, so that both pairs of words are very similar between them.

Measuring **semantic similarity** is very useful for several NLP tasks, such as *Information Retrieval, Text Mining, Machine Translation and evaluation, Summarization, Machine Reading, Deep Question Answering* and many others.

In the next sections we are going to discuss different methods and techniques to estimate the semantic similarity between words, and how to extend these methods to also measure the semantic similarity between sentences.

## 2.2 Word Similarity

It is commonly accepted that there are at least **two kinds of methods** to determine whether two words share some kind of meaning. The first ones are **knowledge-based** word similarity methods, which are based on structured resources such as monolingual or bilingual dictionaries, thesaurus or encyclopedias. Knowledge-bases are very useful because they constitute a highly structured and relevant source of information about words and meanings. Some of the more employed resources of these type are WordNet (Fellbaum 1998) and Wikipedia<sup>1</sup>. Algorithms based on these kind of resources often use the hypernym/hyponym relations (e.g. in WordNet) to compute the semantic between two words. These types of resources are more detailed in Section 2.3.

---

<sup>1</sup><http://www.wikipedia.org>

**Corpus-based** word similarity methods use *large corpora* as a source data for word similarity. The possibility of applying *descriptive approaches* using statistical techniques, having information of the frequency of use, etc. is crucial for extracting important information related to linguistic phenomena. Thus, *unstructured lexical resources* such as monolingual and bilingual corpora provide an additional though less organized source for word similarity. A widely used representation of features in a document (or corpus) is the *Vector Space Model* (VSM) (Salton *et al.* 1975). Corpus-based word similarity methods are presented in Section 2.4.

These techniques are applied at **word level**, and very few at sentence level. This is because *compositionality*, which makes calculating the similarities between sentences very complex and difficult. For instance, the composed meaning of the words 'apple' and 'big' might not be 'large apple', but 'New York'. Compositionality is important as it allows to link 'capital of Switzerland' to 'Berna'. In this thesis we do not explicitly cover compositionality, but it is implicit in the system presented in Chapter 4.

In the next three sections we describe resources and methods of *Knowledge-based* (Section 2.3) and *Corpus-based* (Section 2.4) word similarity, and how to combine them (Section 2.5). Next, in Section 2.6 we describe the most common approaches to measure the similarity for longer snippets of text using word similarity metrics.

## 2.3 Knowledge-based word similarity

In NLP, the use of *on-line dictionaries* or *Machine Readable Dictionaries* (MRDs), a term coined in the 80s referring to dictionaries for human use in digital support, has been studied extensively in the hope that monolingual and bilingual dictionaries might provide a way out of the semantic similarity. Although MRDs are built for human use and they deal with problems such as inconsistencies, too fine-grained ambiguity, circular definitions, etc., MRDs seemed to offer the possibility for enormous savings in time and human effort (Zernik 1991; Briscoe and Boguraev 1989; Wilks *et al.* 1996; Rigau *et al.* 1998).

### 2.3.1 Knowledge-bases and resources

**WordNet**<sup>2</sup> (Miller *et al.* 1991; Fellbaum 1998), is a lexical database for the English language. Its design is inspired by current psycholinguistic and computational theories of human lexical memory. WordNet is by far the most widely-used lexical knowledge base. It contains manually coded information about English nouns, verbs, adjectives and adverbs, and is organized around the notion of *synset*. A synset is a set of words with the same part-of-speech that can be interchanged in a certain context. For example,  $\langle student, pupil, educatee \rangle$  form a synset because they can be used to refer to the same concept. A synset is often further described by a gloss, in the case of the above synset 'a learner who is enrolled in an educational institution', and by explicit semantic relations to other synsets. Each synset represents a concept which is related to other concepts by means of 26 semantic relationships, including hypernymy/hyponymy, meronymy/holonymy, antonymy, entailment, etc (see Section 2.1). Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network meaningfully relates words and concepts, and its structure makes it a useful tool for computational linguistics and natural language processing. It is used in a wide variety of NLP tasks such as Information Extraction (Stevenson and Greenwood 2006), Automatic Summarization (Chaves 2001), Question Answering (Moldovan and Rus 2001), Lexical Expansion (Parapar *et al.* 2005) as a knowledge resource or a dictionary.

WordNet was created and is being maintained at the *Cognitive Science Laboratory* of Princeton University under the direction of psychology professor George A. Miller. Its development began in 1985. Over the years, the project received funding from different government agencies. WordNet is freely and publicly available for download. The actual version of WordNet is 3.1, but this version is available only online. The latest WordNet version for Unix-like systems is 3.0, and contains 82,115 nouns, 13,767 verbs, 18,156 adjectives and 3,621 adverbs, totalling 117,659 synsets. From version 1.5 to 3.0, WordNet has been increased by nearly 26,000 new synsets.

**Wikipedia**<sup>3</sup> is a free online encyclopedia that aims to allow anyone to edit articles. Wikipedia is available in 295 languages. The english version of Wikipedia contains more than 5,363,191 articles, being the largest among the 295 languages.

The content of Wikipedia can be classified into articles, categories, redirections and disambiguation pages. Using this structure we can construct a graph

---

<sup>2</sup><http://wordnet.princeton.edu>

<sup>3</sup><https://www.wikipedia.org>

using the articles and categories as nodes and the links as edges. To gather these links it is necessary search in the text of each article for hyperlink to other articles or categories. Category pages also contain hyperlinks to other category pages, constructing the *category structure*. Additionally, it is also possible to create a dictionary with all the string in Wikipedia. In this dictionary, each entry would contain all possible articles that string could be referencing. These articles can be weighted by their probability to be the actual reference of the entry in the dictionary. (e.g In the dictionary entry for '*Nadal*' the article '*Rafael Nadal*' should have higher probability than '*Lymari Nadal*').

### 2.3.2 WordNet-based methods

One of the most important and popular knowledge-bases is WordNet. This section illustrates some of the best known techniques based on WordNet that allows us to calculate the similarity between words:

- **Path-Length Measure:** This algorithm is based on the principal assumption that the shorter the path between two words is, more similar they are between them.
- **Leacock-Chodorow Measure:** This method is an extension to the Path-Length measure which scales the path length by the depth of the hierarchy, defined as the length of the longest path from a leaf node to the root of the hierarchy ([Leacock and Chodorow 1998](#)).
- **Resnik Similarity Measure:** This algorithm uses the structure of the thesaurus and combines it with probabilistic information extracted from corpora. Resnik's similarity measure supposes that the semantic similarity of two concepts is proportional to the amount of information they share ([Resnik 1995](#)).
- **Lin Similarity Measure:** is an extension the Resnik similarity, introducing the *commonality* and *difference* measures. *Commonality* is a measure that indicates how much two concepts have in common. *Difference* is the measure that indicates that he more differences are between two concepts, the more different they are ([Lin 1997](#)).
- **Jiang-Conrath Distance:** This technique measures unrelatedness between two concepts ([Jiang and Conrath 1997](#)).

- **Hirst-St.Onge Measure:** The algorithm classifies the WordNet relations in three categories: up, down or horizontal. There are also four levels of relatedness: extra strong, strong, medium strong and weak. The extra strong and strong relationship involve words of the same concept (horizontal relation). (Hirst and St-Onge 1998) calculates the score of the relation with the path length between the concepts and the number of changes of direction in that path.

Moreover, (Pedersen *et al.* 2004) created a freely available software package that makes it possible to measure the semantic similarity and relatedness between a pair of concepts (or synsets). It provides six measures of similarity, and three measures of relatedness, all of which are based on WordNet (includes all methods shown above). These measures are implemented as Perl modules called *WordNet::Similarity*<sup>4</sup> which take as input two concepts, and return a numeric value that represents the degree to which they are similar.

### Extended Lesk Measure

The *Lesk Algorithm*<sup>4</sup> (Lesk 1986) is an algorithm based on two assumptions. The first one is that concepts that are nearby between them have more possibilities to share some topic. The second is that related senses can be identified searching overlaps in their glosses.

The algorithm computes simple unigram overlaps in the glosses that are contained in WordNet. The basic idea behind the *Extended Lesk measure* (Patwardhan *et al.* 2003) is that two concepts in a dictionary are similar if they share common words in their glosses. For each common phrase in the glosses of two concepts containing  $n$  words, the Extended Lesk measure assigns a score of  $n^2$ . The total similarity score is the sum of those scores. In addition, Extended Lesk looks for overlap between all glosses of the senses that have a relation (e.g. hypernym, hyponym) with the concepts.

Let  $R$  be the set of possible WordNet relations between two concepts. The Extended Lesk overlap measure is defined as:

$$sim_{eLesk}(c_1, c_2) = \sum_{r, q \in R} overlap(gloss(r(c_1)), gloss(q(c_2))) \quad (2.1)$$

Where  $c_1, c_2$  are two concepts,  $r, q$  are two WordNet relations and  $gloss(r(c))$  is the concatenation of all the senses of  $c$  with relation  $r$ .

---

<sup>4</sup><http://search.cpan.org/dist/WordNet-Similarity/lib/WordNet/Similarity.pm>

### Graph-based Method

This method considers WordNet as a graph  $G = (V, E)$  in which each node represent a concept (synset) or a dictionary word. Each undirected edge represents a relation between synsets and each directed edge represents a link from a dictionary word to a synset. (Hughes and Ramage 2007) presented a random walk algorithm over WordNet, with good results on a similarity dataset. (Agirre *et al.* 2009) improved these results and provided the best results among WordNet-based algorithms on the Wordsim353 dataset.

The method includes two steps. Firstly, it computes a variant of the original PageRank (Lawrence Page *et al.* 1999) called personalised PageRank (T. H. Haveliwala 2002) over WordNet for each word in order to produce a probability distribution over WordNet synsets. Then, it computes the similarity of those words by using the cosine between two vectors created from the probability distributions.

In the first step,  $G$  is considered as a graph with  $N$  vertices  $v_1, \dots, v_N$  and  $d_i$  be the out-degree of node  $i$ ; let  $M$  be a  $N \times N$  transition probability matrix, where  $M_{ji} = \frac{1}{d_i}$  if a link from  $i$  to  $j$  exists, and zero otherwise. Then, the calculation of the *PageRank vector*  $\mathbf{Pr}$  over  $G$  is equivalent to resolving the following equation:

$$\mathbf{Pr} = cM\mathbf{Pr} + (1 - c)\mathbf{v} \quad (2.2)$$

In the equation,  $\mathbf{v}$  is a  $N \times N$  vector whose elements are  $\frac{1}{N}$  and  $c$  is the so called *damping factor*, a scalar value between 0 and 1. The first term of the sum on the equation models the voting scheme described in the beginning of the section. The second term represents, loosely speaking, the probability of a surfer randomly jumping to any node, e.g. without following any paths on the graph. The damping factor, usually set in the [0.85..0.95] range, models the way in which these two terms are combined at each step.

In the second step, once personalized PageRank is computed, it returns a probability distribution over WordNet synsets. The similarity between two words can thus be implemented as the similarity between the probability distributions. Alternatively, we can interpret the probability distribution for a word  $w$  as a vector  $\vec{w}$  of weights  $w_i$  where each dimension  $i$  is a synset, and use the cosine to compute similarity, as in the following equation:

$$\text{sim}(\vec{w}, \vec{v}) = \cos(\vec{w}, \vec{v}) = \frac{\vec{w} \cdot \vec{v}}{\|\vec{w}\| \|\vec{v}\|} \quad (2.3)$$



This method is implemented in the UKB<sup>5</sup> package, a collection of programs for performing graph-based Word Sense Disambiguation and lexical similarity/relatedness using a pre-existing knowledge base (Agirre *et al.* 2009, 2010). UKB has been developed by the IXA<sup>6</sup> group in the University of the Basque Country.

### 2.3.3 Wikipedia-based methods

Lately, a new approach has entered into the scene: building wide coverage knowledge bases from *encyclopedias* developed by Web2.0 communities, such as Wikipedia<sup>7</sup>. Wikipedia is a multilingual, Web-based encyclopedia written collaboratively by volunteers which is available for free. This section describes some methods based on Wikipedia that allows us to calculate the similarity between concepts:

- **WikiRelate!**<sup>8</sup>: This system developed by (Strube and Ponzetto 2006) is based on methods for WordNet (Hirst and St-Onge 1998; Jiang and Conrath 1997; Leacock and Chodorow 1998; Lin 1997; Patwardhan *et al.* 2003; Resnik 1995) and redesigned to work with the Wikipedia. WikiRelate! retrieves all pages from Wikipedia containing the two words for which we want to compute the similarity, and then computes the text overlaps in the content of the articles.
- **Wikipedia Link Vector Model**: This technique is based in the structure of the links and the titles of the Wikipedia articles. The system computes the similarity computing the angle between the vectors of links, weighting them with the probability of each link. This method is explained with more detail in Section 6.2.3.
- **WikiWalk**<sup>9</sup>: WikiWalk (Yeh *et al.* 2009) is a method that uses random walk algorithms on a graph to measure semantic similarity between words. The graph is created by representing each article as a node and each link between articles as an edge. Given two words, WikiWalk uses the Explicit Semantic Analysis (Gabrilovich and Markovitch 2007) to find their corresponding nodes in the Wikipedia graph. After the words are linked to specific nodes,

---

<sup>5</sup><http://ixa2.si.ehu.es/ukb>

<sup>6</sup><http://ixa.si.ehu.es/Ixa>

<sup>7</sup><http://www.wikipedia.org>

<sup>8</sup><http://www.eml-research.de/nlp>

<sup>9</sup><http://wiki-walk-ios.soft112.com/>



semantic similarity is computed by applying personalised Pagerank for each word to create a probability distribution of related nodes. The final score is given by the cosine of the angle between the vectors of their probability distributions.

## 2.4 Corpus-based word similarity

Large corpora has been also used as a source data for semantic similarity. The possibility of applying *descriptive approaches* (those which derive the necessary knowledge from a natural source of data without any pre-existing frame) using statistical techniques, having information of the frequency of use, etc. is crucial for extracting important information related to linguistic phenomena. Thus, *unstructured lexical resources* such as monolingual and bilingual corpora provide an additional though less organized source for semantic similarity.

### 2.4.1 Distributional Semantics

*Distributional Semantics Modelling* (DSM) is an active area of research within the field of natural language processing. In distributional semantics, the meaning of words is explored by looking at their distribution in texts (*You shall know a word by the company it keeps!* (Firth 1957)). The combined contexts of words, represented as feature vectors in a high-dimensional vector space, are indicative of their meanings. These models are named *Vector Space Models*.

#### Weighted word co-occurrence matrices

In VSM the meaning of a content word is represented in terms of a distributed vector, recording its pattern of co-occurrences (sometimes, using specific syntactic relations) with respect other content words within a corpus. Different semantic similarity measures and linguistic phenomena may then be modelled in terms of linear algebra operations (such as cosine) on distributional vectors.

Since distributional semantic models represent words according to their occurrence contexts, they may be used to model word similarity or word association. Two words are similar/related if they co-occur in similar contexts. This idea can be straightforwardly used to acquire pairs (or sets) of related words. We can encode how often a word occurs in a document or in conjunction with another word. Matrices encoding the former are *word-document matrices*, and matrices encoding the latter are *word-word co-occurrence matrices*. A **word-document matrix**

is a  $|D| \times |V|$  matrix, where  $D$  is the collection of documents and  $V$  is the the vocabulary. In these matrices two documents are similar if their vectors are similar, and two words are similar if their vectors are similar. To generate **word-word matrices** windows of certain numbers of words or full paragraphs are used instead of documents. Small windows of 1-3 words encode a more syntactic representation, while longer windows of 4 or more words encode a more semantic representation. These matrices are  $|V| \times |V|$  where  $V$  is the vocabulary, and words are similar if their vectors are similar (they usually co-occur with the same words: context).

Typically *word-word matrices* are very sparse as the majority of words do not appear in conjunction with other words. But there are other word that are very frequent but do not provide much information, such as 'the', 'a', 'from' or 'to'. It is crucial to have a measure of how important or informative is a word. Many studies have used different statistic techniques to measure the significance of terms with respect a corpus text. In Information Retrieval (Baeza-Yates and Ribeiro-Neto 1999; Kageura and Umino 1996; Manning and Schütze 1999) different term-weight measures are used to represent the usefulness of terms in the retrieval process; for example, frequency (Luhn 1957), *signal-to-noise ratio* (Dennis 1964; Salton and McGill 1986), *Pointwise Mutual Information* (PMI) (Church and Hanks 1990), IDF (Jones 1972), relevance weighting methods (Robertson and Jones 1976), and TF-IDF and its variations (Salton and Buckley 1988). Using these measures it is possible to weight word-word co-occurrence matrices to reflect the most salient characteristics.

The final step is to use the co-occurrence matrices to measure the similarity between two words. Most methods to measure the semantic similarity of pairs of words are based on *dot product*, *inner product* or *cosine similarity*. Cosine similarity solves the problem that happens when using *dot product* with words of very different frequency (these vectors are longer). Dividing the *dot product* by the length of the two vectors gives as result the cosine of the angle between both vectors or cosine similarity:

$$\cos\theta = \frac{\vec{w} \cdot \vec{v}}{\|\vec{w}\| \|\vec{v}\|} \quad (2.4)$$

Other methods to compute the similarity of vectors are chi-square statistics (Makoto *et al.* 1976), PMI (Church and Hanks 1990), *Dice coefficient* (Smadja 1993), *log-likelihood ratio* (Dunning 1993) and *Jaccard similarity measure* (Grefenstette 1994).

### Dimensionality reduction

Word co-occurrence matrices are usually very big. If the vocabulary comprises 50,000 words, matrix is going to be an enormous  $50,000 \times 50,000$  matrix. Manipulating matrices of this size is computationally expensive. *Dimensionality reduction* aims to reduce the size of matrices by eliminating highly correlated rows and columns, while maintaining most of the information.

*Singular Value Decomposition* (SVD) is a factorization of a matrix, where for a matrix  $M$  of dimension  $m \times n$  there exists the following factorization:

$$M_{m \times n} = U_{m \times n} S_{n \times n} V_{n \times m}^* \quad (2.5)$$

where  $U$  and  $V$  are orthonormal ( $V^*$  being the conjugate transpose of  $V$ , thus also orthonormal) and  $S$  a diagonal matrix where  $S(1, 1)$  corresponds to the dimension with greatest variability,  $S(2, 2)$  to the second dimension with greatest variability and so on, being  $S(n, n)$  the dimension with least variability.

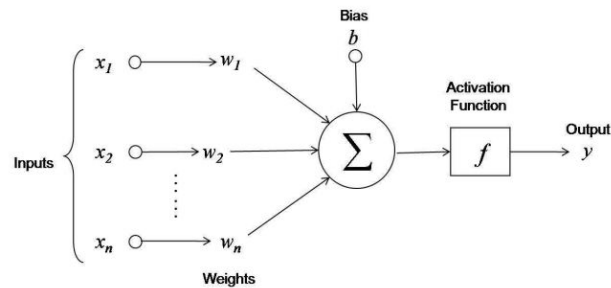
*Latent Semantic Analysis* (LSA) (Deerwester *et al.* 1990; Landauer and Dumais 1997; Schütze 1998) is a technique to derive a  $k$ -dimensional matrix that is an approximation of the original matrix  $M$ . LSA ensures the least information loss for any given value for  $k$ , being a least-squares approximation to the original matrix. In the last two decades LSA have proven to be useful in several NLP tasks. Amongst many others, it have been applied to solve the TOEFL synonym test (Landauer and Dumais 1997; Rapp 2004), automatic thesaurus construction (Schütze 1998), identification of translation equivalents (Rapp 1999), word sense induction and discrimination (Schütze 1998), *Part-of-Speech* induction (Schütze 1995), identification of analogical relations (Turney 2006), PP attachment disambiguation (Pantel and Lin 2000), and semantic classification (Versley 2008).

*Latent Dirichlet Allocation* (LDA) (Blei *et al.* 2003) is another statistical method that learns a set of latent variables, called topics, describing the contents of a document collection. Given a topic model, documents can be viewed as a set of probability distributions over topics,  $\theta$ . The distribution for an individual document  $i$  is denoted as  $\theta_i$ . The similarity between a pair of texts is estimated by comparing their topic distributions (Aletras *et al.* 2012; Aletras and Stevenson 2012). This is achieved by considering each distribution as a vector (consisting of the topics corresponding to an item and its probability) and then computing the cosine of the angle between them. LDA can be used as a dimensionality reduction technique by deriving a probabilistic *word*  $\times$  *topic* VSM.

## 2.4.2 Distributed Representations and Neural Networks

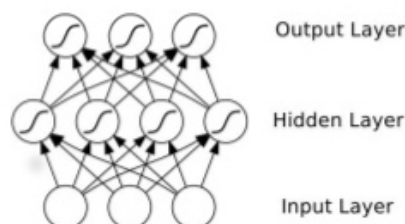
Learning *distributed representations* is a topic very related to *distributional semantics* that has attracted a lot of attention lately. *Distributed representations* of words are vectors that are able to characterize the meaning of that words automatically. Each word corresponds to a vector in an N-dimensional space, where word vectors with a similar meaning are close, and those with different meanings are further away.

A *language model* is a function or algorithm that can be used to learn the statistical characteristics involved in the distribution of certain sequences of words in a naturally written text. Learning this statistic allows us to design a probabilistic model capable of predicting which word comes after a certain sequence of other words.



**Figure 2.1** – Diagram of an artificial neuron. Each neuron is a simple logistic regression  $y = f(Wx + b)$ , where inputs  $x_i$  are weighted by the synaptic weights  $w_i$ , summed up, and passed through the activation function  $f$ , which is usually a sigmoid or  $\tanh$  ( $b$  is a bias term).

*Neural networks* are computational architectures that seeks to mimic the functioning of the human brain. As in the human organ, each (artificial) neuron (Figure 2.1) is connected to other neurons, forming a network (Figure 2.2). Given an input that activates a certain number of neurons, the neural network returns as output the activation of other neurons. Typical neural networks contain an input layer, an output layer, and at least one hidden layer. Modern networks contain from few thousand to a few million neurons, with millions of connections between them. Training these networks has traditionally been very complicated and not very fruitful. This began to change recently thanks to the arrival of *Deep Learning* techniques, which allowed pre-training hidden layers of neural networks (Hinton *et al.* 2006; Erhan *et al.* 2010). Deep Learning made it possible for neural



**Figure 2.2** – A diagram of a neural network with one hidden layer. Image from ‘*Supervised Sequence Labelling with Recurrent Neural Networks*’, PhD dissertation by Alex Graves.

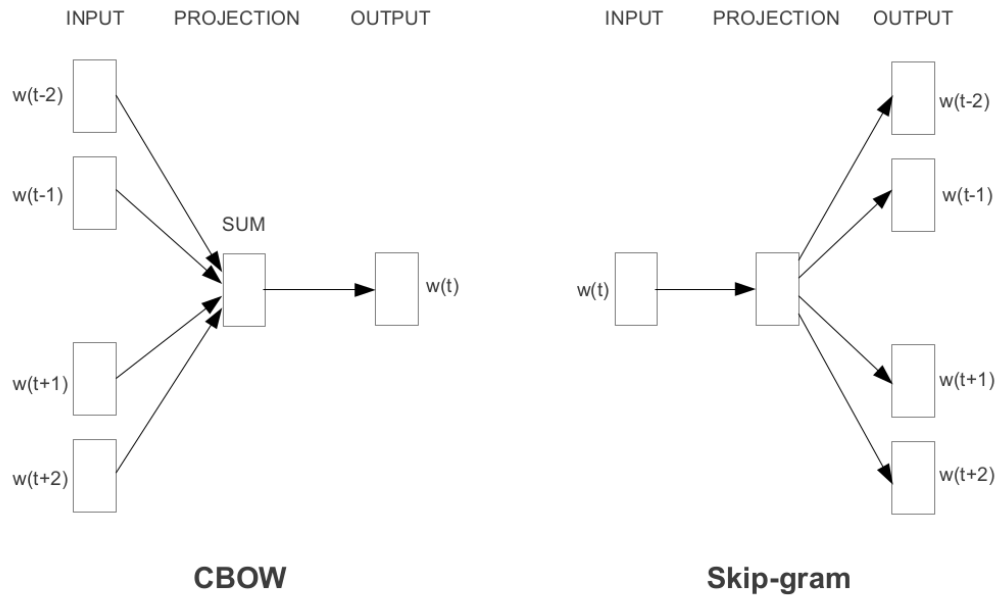
networks to address more complex problems with better results. Neural networks are able to learn distributed representations iteratively, forming an  $N$ -dimensional space where the words have been distributed according to their meaning.

A *neural network language model* (NNLM) is a *language model* based on *neural networks*. (Bengio *et al.* 2003) presents a method that allows to learn distributed representations while at the same time uses those word vectors to predict the probability with which those words occur in that context, reducing the curse of dimensionality. (Collobert and Weston 2008) presented another similar solution for calculating these word vectors optimizing the network via gradient ascent. The derivatives modify the word vectors in a  $L \in \mathbb{R}^{n \times |V|}$  matrix, where  $|V|$  is the size of the vocabulary and  $n$  is the dimensionality. The word vectors inside this matrix capture distributional syntactic and semantic information via the word’s co-occurrence statistics. Once we have learned these word vectors (or embeddings) we can use them to calculate the similarity between words using the cosine similarity between them.

### 2.4.3 Continuous Bag-of-Words and Skip-gram models

(Mikolov *et al.* 2013a) proposed two models for learning distributed representations of words, while minimizing the computational complexity.

The *Continuous Bag-of-Words* (CBOW) model is an architecture similar to a NNLM, but the hidden layer is removed and the projection layer is shared for all words. In this model the order does not affect the projection. In fact, they use a window of four previous words and four following words, where the training objective is to correctly classify the current word. In other words, the model predicts the current word based on its context. The CBOW model architecture is



**Figure 2.3** – On the left, CBOW architecture that predicts the current word based on the context. On the right, Skip-gram architecture that predicts surrounding words based on the current word. Images extracted from [Mikolov et al. 2013a](#).

shown in [Figure 2.3](#).

The *Skip-gram* model is similar to CBOW, but it predicts the context of a given word, instead of predicting the current word based on the context. Using a *hierarchical softmax* function they predict (a window of) words before and after the current word. Increasing this windows improved the resulting word vectors, but also the computational complexity of the model. Following the assumption that distant words are usually less relevant to the current word they lowered the weight of distant word by sampling less of those words. The Skip-gram model architecture is shown in [Figure 2.3](#), on the right side.

Word representations obtained using the *Skip-gram* model were improved in ([Mikolov et al. 2013b](#)) by subsampling words that occur frequently such as 'in', 'the' or 'a'. They also described a simple alternative to the hierarchical softmax called *negative sampling*. Additionally, they present a technique to learn phrase representations by substituting words that appear frequently together in the corpus by tokens (e.g. they substitute 'New York' by 'New\_York').

## 2.5 Combining Knowledge-based and Corpus-based similarity

In general, there are three methods to combine Knowledge-based and Corpus-based similarity. In this section we briefly describe these methods.

The first method is to **use resources to complement others**. Although the vocabulary of WordNet is very extensive, sometimes we are in the case that a give a word is not included in WordNet (or other dictionary). In these cases it is possible to use other resources to complete the missing information. For instance, it is possible to search in corpora, using distributional semantics, words with similar meanings to those words that are not in our dictionary, in order to discover others who are. Once synonyms or words with similar meanings of the target word are detected it is possible to find some of them in our dictionary. (Agirre *et al.* 2009) explored this approach, improving their results.

The second method, and probably the most used one, is to combine knowledge-based and corpus-based similarity using **Machine Learning** (ML) approaches. In this approach similarity scores and other features are computed separately using knowledge and corpus-based techniques. These features and scores are used to feed any machine learning algorithms, such as *Linear Regressors*, *Decision trees* or *Support Vector Regressors*.

The third method proposes to **encode word vectors** using the structure stored in **knowledge bases** (e.g. WordNet or Wikipedia) (Goikoetxea *et al.* 2015). In this model, the meaning of a word is encoded using random walks over the knowledge base, where each random walk generates an *artificial context* for a given word. Then, this artificial context is used to feed a NNLM which is able to learn word vectors. These word vectors differ from the previous two because they encode the meaning based on a structured knowledge base, made by human experts, instead using unlabeled corpora. Wordnet based word vectors<sup>10</sup> provide more precise knowledge, but less coverage.

Recently, (Rychalska *et al.* 2016) used knowledge from WordNet to improve their word representations, with moderate success (see Section 2.8.5). However, the most common solution to combine *Knowledge-based* and *Corpus-based similarity* is to use ML. In this approach the similarities are computed independently using both techniques, then combined using mathematical models able to search patterns and learn a function that can improve results.

---

<sup>10</sup><http://ixa2.si.ehu.es/ukb/>

## 2.6 From Word Similarity to Textual Similarity

All methods presented in Sections 2.3 and 2.4 are constructed/designed to compute the similarity between a pair of words. Now, consider that we want to measure the similarity between these two sentences:

- *The man is smashing garlic.*
- *A man is smashing some garlic.*

Without using any of the methods presented in this chapter we can align words from one sentence to the other: *man-man*, *is-is*, *mashing-smashing*, and *garlic-garlic*. Using a simple **word overlap** metric we can compute a similarity score for these sentences  $S_1$  and  $S_2$ :

$$\text{sim}(S_1, S_2) = \frac{2 * |\text{Aligned}|}{|S_1| + |S_2|} \quad (2.6)$$

where  $\text{Aligned} = \{\text{man} - \text{man}, \text{is} - \text{is}, \text{smashing} - \text{smashing}, \text{garlic} - \text{garlic}\}$  is a set of aligned words between  $S_1$  and  $S_2$ . Following this metric the similarity is 0.8 out of a maximum score of 1, and if stop-words are removed the similarity increases up to 0,86. However, word overlap and other similar measures, like *n-grams comparison*, fail when evaluating sentences with different but similar words, such as the following two:

- *The woman is applying cosmetics to her face.*
- *A girl is putting makeup on her face.*

Although aligning words from one sentence to another is not very difficult for humans, it is not trivial for computers. Below, we describe two techniques to compute the similarity between sentences using the methods and resources previously presented in this chapter.

### Sentence Similarity by Pairwise Word Similarity

(Mihalcea *et al.* 2006) presented a method for measuring the semantic similarity of texts as a function of the semantic similarity of the components words. They did this by combining metrics of word-to-word similarity (similarity scores for a given word pair) and word specificity (IDF scores for the words, see below) in a



## 2.6. FROM WORD SIMILARITY TO TEXTUAL SIMILARITY

---

formula capable to assign a good semantic similarity score for to input sentences  $S_1$  and  $S_2$ :

$$sim(S_1, S_2) = \frac{1}{2} \left( \frac{\sum_{w_i \in S_1} (idf(w_i) * \max_{w_j \in S_2} Sim(w_i, w_j))}{\sum_{w_i \in S_1} idf(w_i)} + \frac{\sum_{w_j \in S_2} (idf(w_j) * \max_{w_i \in S_1} Sim(w_j, w_i))}{\sum_{w_j \in S_2} idf(w_j)} \right) \quad (2.7)$$

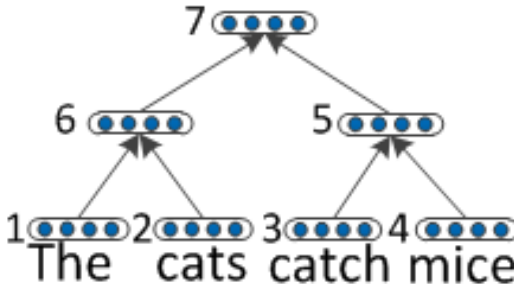
where *Inverse Document Frequency* (IDF) is an inverse function of the number of documents in which that term occurs that can be used to quantify the specificity of a term/word. Taking into account the specificity of words allows to this formula assigning higher weight to a semantic matching identified between two specific words (e.g *collie* and *sheepdog*), and give less importance to the similarity measured between generic concepts (e.g *get* and *become*). It is possible to substitute the IDF for any other word specificity metric, and  $sim(w_i, w_j)$  can be any metric to measure the similarity between two words.

The formula aligns each word  $w_i$  from sentence  $S_1$  to the word  $w_j$  in sentence  $S_2$  with the highest semantic similarity (and vice versa). In this process only words with the same part-of-speech (PoS) are considered. In other words, when aligning a *noun* all words with other PoS are ignored (for instance, it cannot be aligned to a *verb*). Furthermore, if two words are identical in both sentences, their similarity is going to be 1.

### Autoencoders

Architectures shown in Section 2.4.1 are very useful for representing words, but they are not capable of representing more complex constructions, such as *phrases* or *sentences*. *Recursive Autoencoders* (RAE) (Pollack 1990; Socher et al. 2010, 2011b) architectures allow to learn semantic vector representations of phrases or sentences. First, sentences are parsed to generate dependency trees. Then, RAE uses the word vectors of word in leaves to recursively compute the representations of intermediate nodes (or subtrees), obtaining a final vector that characterizes the meaning of the full phrase (or sentence). Figure 2.4 shows an example of how a RAE encodes the distributed representation of a sentences.

Once these sentence representations are generated it is possible to compute the similarity between sentences in the same way as with word representations, by using the cosine similarity of vectors.



**Figure 2.4** – Example of a RAE encoding the distributional representation of a full sentence, extracted from (Socher *et al.* 2011a). Vectors 1-4 are the distributional representations for the words *The*, *cats*, *catch* and *mice*, respectively. A RAE is first applied for each intermediate node in the dependency tree, computing phrase representations for *The cats* (vector 6) and *catch mice* (vector 5), and then to vectors 5 and 6 to compute the representation of the full sentence, *The cats catch mice* (vector 7).

## 2.7 Evaluation

To evaluate systems for semantic similarity it is necessary to have datasets annotated with appropriate values for each pair of words or sentences. Typically, these datasets are created using annotations assigned by humans. Datasets for semantic similarity were scarce when this thesis was proposed. However, there were at least two datasets for word similarity, and other two for longer snippets of text. In this section we present those datasets for semantic similarity.

The first one, **RG** dataset, consists of 65 pairs of words collected by (Rubenstein and Goodenough 1965), who had them judged by 51 human subjects in a scale from 0.0 to 4.0 according to their similarity, but ignoring any other possible semantic relationships that might appear between the terms. This dataset is a consequence of a study about the relationship between similarity of context and similarity of meaning (synonymy). Rubenstein and Goodenough asked to humans how the proportion of words common to context containing a word *A* and to the contexts containing a word *B* was related to the degree to which *A* and *B* were similar in meaning. These method assume that pairs of words which have many contexts in common are semantically closely related. Using 65 pairs of words (which range from highly synonymous pairs to semantically unrelated pairs) the relation is shown between similarity of meaning The 65 word pairs consist of ordinary English words.

The second dataset, **WordSim-353** (Finkelstein *et al.* 2002) contains 353 word

pairs, each associated with an average of 13 to 16 human judgements. In this case, both similarity and relatedness are annotated without any distinction. Several studies indicate that the human scores consistently have very high correlations with each other (Miller *et al.* 1991; Resnik 1995), thus validating the use of these datasets for evaluating semantic similarity. The dataset contains two sets of English word pairs along with human-assigned similarity judgements. The collection can be used to train and/or test computer algorithms implementing semantic similarity measures. The first set (set1) contains 153 word pairs along with their similarity scores assigned by 13 subjects. The second set (set2) contains 200 word pairs, with their similarity assessed by 16 subjects. Subjects' names have been replaced by ordinal numbers (1..13, or 1..16) to protect their privacy; identical numbers in the two sets do not necessarily correspond to the same individual. Each set provides the raw scores assigned by each subject, as well as the mean score for each word pair. For convenience, there is a combined set (combined) that contains a list of all 353 words, along with their mean similarity scores. The combined set is merely a concatenation of the two smaller sets. (Agirre *et al.* 2009) also proposed to split the WordSimilarity-353 collection into two datasets, one focused on measuring similarity, and the other one on relatedness.

These two datasets are word similarity datasets, and they are not important for this research. Regarding text similarity, there were two similarity datasets, (Li *et al.* 2006) and (Lee *et al.* 2005). These datasets include dictionary definitions and news documents, respectively. Thus, they are datasets for similarity **above word level**: sentence similarity and document similarity.

The **Li** dataset includes 65 sentence pairs, which correspond to the *dictionary definitions* for the 65 word pairs in the *RG dataset*. Where more than one sense of a word was given in the dictionary, the first noun sense in the list was chosen. The authors asked human informants to assess the meaning of the sentence pairs on a scale from 0.0 (minimum similarity) to 4.0 (maximum similarity). Each sentence pair was presented on a separate sheet, and the order of presentation of the sentence pairs was randomized to avoid biases due to the order of presentation. They assigned a semantic similarity score calculated as the mean of all annotations to each of the 65 sentence pairs. The distribution of scores was very skewed towards low similarity values. While the dataset is very relevant to semantic sentence similarity, it is too small to train, develop and test typical machine learning based systems.

The **Lee** dataset comprises 50 documents from the *Australian Broadcasting Corporation's* news mail service, ranging in length from 51 to 126 words, covering various topics. Subjects were asked to judge the similarity of document pairs

on a five-point scale (with 1.0 indicating 'highly unrelated' and 5.0 indicating 'highly related'). Each possible pair of documents (excluding self-comparisons) was presented between eight and twelve times, in a random order, and which documents were shown on the left and which ones on the right was also randomly determined. As in the previous dataset, the distribution of scores was heavily skewed towards low similarity values. The semantic similarity score was calculated as the mean of all annotations for each document pair, and then normalized to a 0-1 scale. This second dataset comprises a larger number of document pairs, but it goes beyond sentence similarity into document similarity.

Currently there are other datasets related with textual similarity, out of the scope of this thesis, but based on the datasets created, annotated and presented in this thesis. Although they did not exist when this research began they are listed here for completeness. *Sentences Involving Compositional Knowledge (SICK)* consists of about 10,000 English sentence pairs annotated for relatedness in meaning and entailment (Bentivogli *et al.* 2016). It is derived from caption of images from 8K ImageFlickr (Hodosh *et al.* 2013) dataset and from our MSRvideo dataset from 2012 (see Section 3.3.1). The sentences have been simplified to make them easier to process by compositional model, and some of these sentences were modified and transformed to create variants. *Paraphrase and Semantic Similarity In Twitter (PIT-2015)* is a dataset constructed crawling Twitter's trending topics and their associated tweets (Xu *et al.* 2015). It contains 17,790 sentence pairs for training set, 972 sentence pairs for development set, and 972 sentence pairs for testing set, all of them annotated for paraphrase and semantic similarity. Scores for semantic similarity were annotated following the procedure introduced in Chapter 3).

Although *Textual Entailment (TE)* is not directly textual similarity, it is a related task. The *Stanford Natural Language Inference (SNLI)* corpus is a collection of 570k human-written English sentence pairs manually labelled for TE (Bowman *et al.* 2015). Sentence pairs are labelled with *entailment*, *contradiction*, and *neutral*, and they serve for evaluation and also to develop NLP systems.

## 2.8 Best systems for Semantic Textual Similarity

In this section we describe the best and more important systems for *Semantic Textual Similarity (STS)* task, which is going to be introduced in the next chapter. These systems make use of the methods presented in Sections 2.3, 2.4, 2.5 and 2.6 to compute the similarity between two sentences. They are presented in chronological order, and they were the *best system* in the year they were presented in

at least one of the datasets. These systems are referenced later, comparing them with the system presented in Chapter 4.

### 2.8.1 DKPro

**DKPro** (Bar *et al.* 2012) is the system which ranked first in the 2012 edition of the STS task. It uses a simple log-linear regression model, trained on the training data, to combine multiple text similarity measures of varying complexity. They generate simple string-based features, but also more complex *Semantic Similarity* measures. The final models uses 20 features to feed this linear regressor, out of the possible 300+ features implemented, and they train a different regressor for each of the datasets.

The **String-based** features include the *Longest Common Substring* (Gusfield, 1997), the *Longest Common Subsequence* (Allison and Dix, 1986) and *Greedy String Tiling* (Wise, 1996). In addition to the previous features they also incorporate *n-grams* comparison using the (Barrón-Cedeño *et al.*, 2010) implementation and the *Jaccard coefficient* following (Lyon *et al.*, 2001), and the *containment measure* (Broder, 1997).

As **Semantic Similarity** measures, it uses graph-based representation of words and the semantic relations between them, running the algorithms from (Jiang and Conrath, 1997), (Lin, 1998b) and (Resnik, 1995) on WordNet (Fellbaum, 1998). To scale the pairwise word similarity to the sentences level they used the aggregation method presented in (Mihalcea *et al.* 2006). They also used *Explicit Semantic Analysis* (ESA) (Gabrilovich and Markovitch, 2007) on WordNet, Wikipedia and Wiktionary<sup>11</sup>. Similarity scores from a *Distributional Thesaurus* (Lin 1998a), computed on 10M dependency-parsed sentences of English newswire as a source for pairwise word similarity, were also used to feed the linear regressor, but only a feature based on cardinal number was selected.

**Text expansion** methods such as a *Lexical Substitution System* (Biemann, 2013) based on supervised word sense disambiguation, and the *Moses SMT system* (Koehn *et al.*, 2007 for statistical machine translation were also used.

Additional measures related to structure and style were generated, including computing *stopword n-grams* (Stamatatos, 2011), *part-of-speech n-grams*, *word-*

---

<sup>11</sup>**Wiktionary** (<https://www.wiktionary.org/>) is a multilingual, web-based project to create a free content dictionary of all words in all languages. The main goal of Wiktionary is to provide an instrument to help in the understanding of the words, and not only their definitions. It contains synonyms, antonyms, translations to other languages, etymologies and more. The content of Wiktionary can be used in the same way as Wikipedia.

*pair order* and *word-pair distance* (Hatzivassiloglou *et al.*, 1999), a function for *word frequencies* (Dinu and Popescu, 2009), and statistical properties such as *type-token ratio* (TTR) (Templin, 1957) and *sequential TTR* (McCarthy and Jarvis, 2010).

### 2.8.2 Takelab

**TakeLab** (Šarić *et al.* 2012) is the system which ranked second in the 2012 edition of the STS task. However it was the best system on two of the datasets, *MSRpar* and *MSRvid*. The system uses a *Support Vector Regression* (SVR) model trained on multiple features from word and semantic similarity measures.

**Word similarity** measures include both knowledge-base and corpus-based approaches. The former measures are based on WordNet: *lowest common subsumer* (LCS), *PathLen similarity* (Bird, 2006) computed using the *NLTK library* (Leacock and Chodorow, 1998) and *Lin similarity* (Lin, 1998b). The latter measures are computed using LSA over the *New York Times Annotated Corpus* (NYT) (Sandhaus 2008) and Wikipedia.

**Semantic Similarity** measures include *n-gram overlap features* and *WordNet-Augmented Word Overlap*, a mechanism to assign partial scores to words that are not common to both sentences, *syntactic features* such as *Syntactic Roles similarity* (Oliva *et al.*, 2011) and *Syntactic Dependencies Overlap* similar to the proposal on (Robert and Paris, 2006), and additional features such as *sentence length differences*, *numbers overlap* and *named entity features*.

### 2.8.3 Ebiquity-Core

**Ebiquity-Core** (Han *et al.* 2013) is the best system by mean correlation in the 2013 edition of STS task. It combines LSA word similarity and WordNet knowledge, using an align-and-penalize approach and SVR models.

As **Word similarity** measure they performed LSA on the Web Corpus from the Stanford WebBase project<sup>12</sup>. They employed SVD to improve their word to word similarity scores. Then they used WordNet to increase the similarity between two words if certain conditions were met, such as two words being in the same WordNet synset or one word is the direct hypernym of the other. With this method they were able to increase the similarity between words like *doctor* and *hospital*.

To assign a final STS score they used an align-and-penalize approach, which was defined as follows:

---

<sup>12</sup><http://bit.ly/WebBase>

$$STS = T - P' - P'' \quad (2.8)$$

where  $T$  is the term alignment score and  $P'$  is the penalty for bad alignments.  $P''$  is supposed to be a penalty for syntactic contradictions, but they did not implement this penalty and the final submission for the task did not use this penalty. An alignment was considered as bad if the score of that was lower than 0.05, or if the terms were antonyms according to a collection of antonyms extracted from WordNet (Mohammad *et al.* 2008).

The team submitted two other systems using a SVR models trained on 52 features, but the results of these two systems were worse than the align-and-penalize approach.

#### 2.8.4 DLS@CU

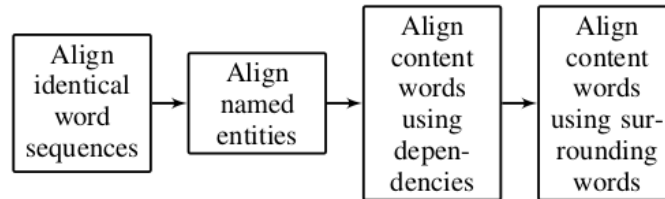
**DLS@CU** (Sultan *et al.* 2014b, 2015) is the system that ranked first in the 2014 and 2015 edition of the STS task.

The system presented in 2014 makes use of a word aligner that aligns related words based on if they are semantically similar or they occur in similar semantic contexts. The proportion of aligned content words between both sentences can be used to assign a similarity score.

This alignment process is applied in a four step pipeline, as shown in Figure 2.5. In the first step the system aligns word sequences that are identical in both sentences and that contains *at least one content word*. In the second step they align named entities using the *Stanford Named Entity Recognizer* (Finkel *et al.* 2005). In the third step they align content words using a dependency-based contextual similarity that defines the context of words using the syntactic dependencies. If they can match the context of the words as equivalent, using the table presented in (Sultan *et al.* 2014a), they align the words. In the last step they align content words using a window of 3 words to the right and 3 words to the left. In their best run, for the OnWN dataset, they used a stop-word list to improve results.

The final STS score is a function of the proportions of the aligned content words in the two sentences. Once they compute the proportion of aligned content word in both sentences, they use the *harmonic mean* of both proportion values to calculate the final score. They submitted this system to the 2015 competition with a minor change in the computation of the proportions of the aligned content words. In the improved system a single proportion is computed, summing the number of aligned content words in both sentences, and dividing it by the total number of words in both sentences. They did other changes, like using the *The*





**Figure 2.5** – The alignment pipeline of the DLS@CU system.

*Paraphrase Database* (PPDB) (Ganitkevitch *et al.* 2013) to identify semantically similar words (helping in the alignment process), or using the *Levenshtein distance*<sup>13</sup> of 1 to detect misspellings. This improved system ranked 5th in 2015 by mean correlation, 0.96 points below their new system.

The system that ranked 1st in 2015 is a combination of the previous system (the improved system from 2014) and similarities from compositional word vectors. Using 400-dimensional word vectors from (Baroni *et al.* 2014b) they constructed sentences vectors by combining the word vectors. To do so they compute the centroid of the word vectors of the lemmas of the content words in the sentence. Once they have the two sentence vectors they compute the similarity between them using the cosine similarity. The combination of these two preliminary STS scores is performed using a ridge regression model as implemented in (Pedregosa *et al.* 2011).

### 2.8.5 Samsung-Ensemble

**Samsung-Ensemble** (Rychalska *et al.* 2016) is the best system by mean correlation in the 2016 edition of STS task. It uses an ensemble classifier to perform *Linear Support Vector Regression* (LSVR) (Drucker *et al.* 1997) over three other classifier: a base word aligner (Sultan *et al.* 2015), a bi-directional *Gated Recurrent Neural Network* (GRNN) (Cho *et al.* 2014; Chung *et al.* 2014) and a third classifier that is a combination of RAE and a WordNet based *award-penalty* system using a *Support Vector Machine* (SVM).

The word aligner is the same used in the system that won the 2015 edition of STS (Section 2.8.4), with two small variations to handle negations and antonyms. If there is a negation in only one sentence, the score is reduced to 0. To detect antonyms they use a list derived from WordNet, and if they found one, the score

<sup>13</sup>The Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.



is also reduced to 0. This *modified aligner* is then used to feed a LSVR, altogether with other features, to generate a *corrected aligner*.

The third classifier is a RAE for unsupervised training of sentence vectors that uses a WordNet base approach to improve the performance. RAE uses parse trees and the word vectors provided by (Pennington *et al.* 2014) to construct a vector that can represent the meaning of the sentence. The module *awards* the pairs of words with positive semantic similarity and *penalizes* the out-of-context words and disjoint similar contexts. The WordNet based system also helps in this process by adjusting the *Euclidean distance* of the word vectors of the leaves in the dependency tree. For instance, if they have two leaves with the words 'woman' and 'lady', the word vectors A and B respectively, and a WordNet similarity  $\epsilon$  they refine one of the word vectors with the following formula:

$$A_{refined} = \epsilon A + (1 - \epsilon)B \quad (2.9)$$

These refined vectors are used to reconstruct the intermediate nodes of the dependency tree, and the final sentence representation, using RAE. The subtrees generated in this step are used to fill a distance matrix using the Euclidean distances, which are modified by the WordNet award-penalize strategy (this modification is independent to the word vector refinement process we have seen above). After adjusting this numbers to fall in the 0-5 range, the systems performs dynamic pooling as seen in (Socher *et al.* 2011a) to be able to compare sentences of different lengths. These matrices, along with additional 12 features were used to feed a SVM.

In order to compute the final STS they trained a LSVR using the two aligners, the RAE with WordNet features, and the GRNN with the output neural network described in (Tai *et al.* 2015).

## 2.9 Conclusions

This chapter has reviewed the state of the art in the area of semantic textual similarity. The concepts of word similarity and relatedness have been defined, and different methods that can be used to compute similarity have been presented. We have also reviewed the datasets that were available for semantic similarity, explaining how they were constructed. The best systems that make use of these techniques and datasets have also been described.

As we have seen, datasets for semantic similarity used to be very small. This makes them difficult to use for the majority of presented techniques, since most

## CHAPTER 2. BACKGROUND

---

of them require large amounts of data to find the most salient characteristics.

In the next chapter we present a new task, *Semantic Textual Similarity*. This task extends the semantic similarity presented in this chapter. In addition to the definition of the task, in the next chapter we also present the datasets for STS, the largest set of data for semantic similarity.

## Semantic Textual Similarity

This chapter describes the contributions made in the design and organization of the *Semantic Textual Similarity* (STS) task. This work has involved designing the task and supporting the creation of datasets for similarity tasks, as well as the organization of the task itself. After the introductory section, in the Section 3.2 we detail and explain how the task was designed. Next, Section 3.3 presents the various sources of the STS data and Section 3.4 presents the annotation procedure used. Then, Section 3.5 investigates the evaluation of STS systems, analyses the attraction of the task over the year, presents some of the best systems that participated in the task and summarizes the tools and resources used by participants. Finally, Section 3.6 draws some conclusions.

### 3.1 Introduction

STS aims to measure the degree of semantic equivalence between two sentences. The objective was to define a task that could assign graded similarity values. This graded similarity should intuitively capture the notion of intermediate shades of similarity, such as pairs of text that differ only in minor nuanced aspects of meaning, in relatively important differences, down to pairs that share only some details or that only have in common being about the same topic. For example, consider these two sentences:

- *The woman is playing the violin.*
- *The young lady enjoys listening to the guitar.*

Undoubtedly, these two sentences are not equivalent, but it is easy to note that they are somehow similar. Both sentences are on the same topic, as they are describing two actions (play/listen) related with musical instruments. Realizing that 'violin' is similar to 'guitar' and that 'woman' is also similar to 'young lady' is an easy task for humans. Now, consider these two other sentences:

- *In May 2010, the troops attempted to invade Kabul.*
- *The US army invaded Kabul on May 7th last year, 2010.*

These two sentences are also similar, and it is not hard to assure that they are more similar than the previous two. To do this we must look at why the sentences of the two examples are different. In the first example, although they share the same topic, they are **different actions** performed by **different persons**. However, in the second example there are **small details** that make them differ. The first sentence does not detail the nationality of the troops, although a person with knowledge about the war of Afghanistan could easily establish this relation. Moreover, the first sentence says that the troops 'attempted' to invade Kabul, and in the second it is implied that this invasion culminated with success. This distinction is also simple to qualify for someone who is informed about this war. A person can add what is missing in the sentences (relatively) easily, or ignore the differences without changing too much the meaning (as in the case of the date). Therefore, both sentences are similar but not equivalent, and it is not difficult to evaluate that this pair of sentences is more similar than the first example.

### 3.2 Design of the STS task

STS aims to establish a unified framework for the evaluation and measurement of the degree of semantic similarity between sentences. As we have seen, measuring the degree of similarity is not straightforward. To make this task easier, instructions were created to define each of these ranges. Six degrees of similarity were designed, ranging from 0 (sentences are on different topics) to 6 (sentences are completely equivalent). In order to facilitate this task, we assigned example sentences for each of the scores. The final instructions and examples for STS are shown in Figure 3.1.

There are other tasks that are related to STS, and we have been inspired by some of them. In the next section we talk about these tasks, and their relation and differences with STS.

- (5) The two sentences are completely equivalent, as they mean the same thing.  
*The bird is bathing in the sink.*  
*Birdie is washing itself in the water basin.*
- (4) The two sentences are mostly equivalent, but some unimportant details differ.  
*In May 2010, the troops attempted to invade Kabul.*  
*The US army invaded Kabul on May 7th last year, 2010.*
- (3) The two sentences are roughly equivalent, but some important information differs/missing.  
*John said he is considered a witness but not a suspect.*  
*"He is not a suspect anymore." John said.*
- (2) The two sentences are not equivalent, but share some details.  
*They flew out of the nest in groups.*  
*They flew into the nest together.*
- (1) The two sentences are not equivalent, but are on the same topic.  
*The woman is playing the violin.*  
*The young lady enjoys listening to the guitar.*
- (0) The two sentences are on different topics.  
*John went horse back riding at dawn with a whole group of friends.*  
*Sunrise at dawn is a magnificent view to take in if you wake up early enough for it.*

**Figure 3.1** – STS annotation scores with explanations and examples.

### 3.2.1 Related datasets and tasks

The first step in the designing process of the STS task was to consult similar sources for inspiration. We decided to investigate if we can reuse collections of existing datasets from tasks that are related to STS. In Section 2.7 we have seen two datasets for semantic similarity, Li and Lee. Both datasets are interesting, but the first is very small, including only dictionary definitions, and the second is a dataset that deals exclusively with similarity between documents.

*Textual Entailment* (TE) (Dagan et al. 2010) and *Paraphrase detection* (PARA) (Dolan and Brockett 2005) are two of the tasks that are most similar to STS. STS differs from TE in as much as it assumes symmetric graded equivalence between the pair of textual snippets. In the case of TE the equivalence is directional (e.g. a car is a vehicle, but a vehicle is not necessarily a car). Additionally, STS differs

from both TE and PARA in that, rather than being a binary yes/no decision (e.g. a vehicle is not a car), STS wants to incorporate the notion of graded semantic similarity (e.g. a vehicle and a car are more similar than a wave and a car).

However, since they also have certain similarities, we studied the pairs of text from the RTE challenge (Dagan *et al.* 2006). The first editions of the challenge included pairs of sentences as the followings:

T: The Christian Science Monitor named a US journalist kidnapped in Iraq as freelancer Jill Carroll.

H: Jill Carroll was abducted in Iraq.

In TE there is a text (first sentence) and a hypothesis (second sentence), being usually the hypothesis a shorter phrase than the text. Although these pairs of text are interesting we decided to discard them from this task because the length of the hypothesis was typically much shorter than the text, and we did not want to bias the STS task in this respect.

In the next section we describe how we collected the sentences to create our datasets for STS. Next, we describe the annotation process in Section 3.4.

### 3.3 Gathering datasets for STS

The main goal of this thesis is to design the *Semantic Textual Similarity* task, but also to create and annotate datasets for it. To construct these datasets the first step was to gather naturally occurring pairs of sentences with different degrees of semantic equivalence. This was a challenge in itself as if we took pairs of sentences at random, the vast majority of them would be totally unrelated, and only a very small fragment would show some sort of semantic equivalence. When we started the task in 2012 we observed that there was no comparable existing dataset extensively annotated for pairwise semantic sentence similarity. We approached the construction of the first STS datasets with the goal of gathering a substantial amount of sentence pairs from diverse datasets. Moreover, we investigated reusing a collection of existing datasets from tasks that are related to STS, as seen in Section 3.2.1. Although we discarded the dataset from TE for the length difference between the text and the hypothesis, we found that the datasets from PARA such as the *Microsoft Research Paraphrase Corpus* (MSRP) could be very useful.

Next we will describe each of these sources and detail how the sentences were selected in each of the years. See Table 3.1 for the number of selected pairs per dataset.

### 3.3. GATHERING DATASETS FOR STS

Year	Dataset	Pairs	Source
2012	MSRpar	1500	newswire
2012	MSRvid	1500	videos
2012	OnWN	750	glosses
2012	SMTeuroparl	1500	WMT eval.
2012	SMTnews	750	WMT eval.
2013	FNWN	189	glosses
2013	HDL	750	newswire
2013	OnWN	561	glosses
2013	SMT	750	MT eval.
2014	Deft-forum	450	forum posts
2014	Deft-news	300	news summary
2014	HDL	750	newswire headlines
2014	Images	750	image descriptions
2014	OnWN	750	glosses
2014	Tweet-news	750	tweet-news pairs
2015	Ans.-forum	375	Q&A forum answers
2015	Ans.-student	750	student answers
2015	Belief	375	committed belief
2015	HDL	750	newswire headlines
2015	Images	750	image descriptions
2016	Ans.-Ans.	254	Q&A forum answers
2016	HDL	249	newswire headlines
2016	Plagiarism	230	short-answer plag.
2016	Postediting	244	MT postedits
2016	Quest.-Quest.	209	Q&A forum questions

**Table 3.1** – Summary of 2012, 2013, 2014, 2015 and 2016 datasets.

#### 3.3.1 STS 2012 datasets

The first edition of STS was held in 2012 as part of the *Semantic Evaluation* (SemEval) series of workshops (Agirre *et al.* 2012). For this competition we created five different datasets, three of them as train data, with their respective test data, and two additional 'surprise' test sets without training data. The train data was composed of pairs of *Microsoft Research Paraphrase* (**MSRpar**), *MSR Video Paraphrase Corpus* (**MSRvid**) and *Statistical Machine Translation from Europarl* (**SMT-Europarl**). The test data was composed with additional sentences from the same sources and with the same number of pairs, and the two surprise dataset with

sentences from Statistical Machine Translation from news (**SMTnews**), and pairs of glosses (**OnWN**).

Microsoft Research (MSR) has pioneered the acquisition of paraphrases with two manually annotated datasets. The first, called MSR Paraphrase (**MSRpar** for short) has been widely used to evaluate text similarity algorithms. It contains 5801 pairs of sentences gleaned over a period of 18 months from thousands of news sources on the web (Dolan *et al.* 2004). 67% of the pairs were tagged as paraphrases. The inter annotator agreement is between 82% and 84%. Complete meaning equivalence is not required, and the annotation guidelines allowed for some relaxation. The pairs which were annotated as not being paraphrases ranged from completely unrelated semantically, to partially overlapping, to those that were almost-but-not-quite semantically equivalent. In this sense our graded annotations enrich the dataset with more nuanced tags, as we will see in the following section. We followed the original split of 70% for training and 30% for testing. A sample pair from the dataset follows:

- The Senate Select Committee on Intelligence is preparing a blistering report on prewar intelligence on Iraq.
- American intelligence leading up to the war on Iraq will be criticized by a powerful US Congressional committee due to report soon, officials said today.

In order to construct a dataset which would reflect a uniform distribution of similarity ranges, we sampled the MSRpar dataset at certain ranks of string similarity. We used the implementation readily accessible at CPAN<sup>1</sup> of a well-known metric (E. Ukkonen 1985). We sampled equal numbers of pairs from five bands of similarity in the [0.4 .. 0.8] range separately from the paraphrase and non-paraphrase pairs. We sampled 1500 pairs overall, which we split 50% for training and 50% for testing.

The second dataset from MSR corpus is **MSRvid**. The authors showed brief video segments to annotators from *Amazon Mechanical Turk* (AMT) and were asked to provide a one-sentence description of the main action or event in the video (David L. Chen and Dolan 2011). Nearly 120 thousand sentences were collected for 2000 videos. The sentences can be taken to be roughly parallel descriptions, and they included sentences for many languages. The sampling procedure from this dataset is similar to that for MSRpar. We construct two bags of data to draw samples. The first includes all possible pairs for the same video, and the

---

<sup>1</sup><http://search.cpan.org/~mlehmman/String-Similarity-1.04/Similarity.pm>



### 3.3. GATHERING DATASETS FOR STS

second includes pairs taken from different videos. Figure 3.2 shows a video and corresponding descriptions.



- A person is slicing a cucumber into pieces.
- A chef is slicing a vegetable.
- A person is slicing a cucumber.
- A woman is slicing vegetables.
- A woman is slicing a cucumber.
- A person is slicing cucumber with a knife.
- A person cuts up a piece of cucumber.
- A man is slicing cucumber.
- A man cutting zucchini.

**Figure 3.2** – Video and corresponding descriptions from MSRvid

Note that not all sentences from the same video were equivalent, as some descriptions were contradictory or unrelated. Conversely, not all sentences coming from different videos were necessarily unrelated, as many videos were on similar topics. We took an equal number of samples from each of these two sets, in an attempt to provide a balanced dataset between equivalent and non-equivalent pairs. The sampling was also done according to string similarity, but in four bands in the  $[0.5 .. 0.8]$  range, as sentences from the same video had a usually higher string similarity than those in the MSRpar dataset. We sampled 1500 pairs overall, which we split 50% for training and 50% for testing. A sample pair from the dataset follows:

- The man is seasoning the sausages.
- The man added seasoning to water in a bowl.

To construct the **SMT-Europarl** dataset, given the strong connection between STS systems and Machine Translation evaluation metrics, we sampled pairs of segments that had been part of human evaluation exercises. Those pairs included a reference translation and a automatic Machine Translation system submission, as follows:

## CHAPTER 3. SEMANTIC TEXTUAL SIMILARITY

---

- The only instance in which no tax is levied is when the supplier is in a non-EU country and the recipient is in a Member State of the EU.
- The only case for which no tax is still perceived "is an example of supply in the European Community from a third country.

We selected pairs from the translation shared task of the 2007 and 2008 *ACL Workshops on Statistical Machine Translation* (WMT) (Callison-Burch *et al.* 2007, 2008). For consistency, we only used French to English system submissions. The training data includes all of the Europarl human ranked fr-en system submissions from WMT 2007, with each machine translation being paired with the correct reference translation. The test data is comprised of all Europarl human evaluated fr-en pairs from WMT 2008 that contain 16 white space delimited tokens or less.

In addition, we created another dataset (**SMTnews**) comprising all the human ranked fr-en system submissions from the WMT 2007 news conversation test set. This dataset was used as test data, without train data. A sample pair from the dataset follows:

- This gross error is leading Russia to political ruin.
- And this gross mistake is conducting Russia policy to his downfall.

The next set was radically different as it comprised pairs of glosses from OntoNotes 4.0 (Hovy *et al.* 2006) and WordNet 3.1 (Christiane Fellbaum 1998) senses (**OnWN**). The mapping of the senses of both resources comprised 110K sense pairs. The similarity between the sense pairs was generated using simple word overlap. 50% of the pairs were sampled from senses which were deemed as equivalent senses, the rest from senses which did not map to one another. A sample pair from the dataset follows:

- a short lyric or poem intended to be sung
- a narrative song with a recurrent refrain.

### 3.3.2 STS 2013 datasets

In the competition of 2013 (Agirre *et al.* 2013c) we did not provide training data, but it was established that the datasets from the last year would be used as training. This rule would apply for the subsequent editions of the competition, reusing

### 3.3. GATHERING DATASETS FOR STS

---

the datasets from previous years as training. The new dataset released in 2013 comprises pairs from news headlines (**HDL**), MT evaluation sentences (**SMT**) and pairs of glosses (**OnWN** and **FNWN**).

For **HDL**, we used naturally occurring news headlines gathered by the *Europe Media Monitor* (EMM) engine (Best *et al.* 2005) from several different news sources. EMM clusters together related news. Our goal was to generate a balanced dataset across the different similarity ranges, hence we built two sets of headline pairs: (i) a set where the pairs come from the same EMM cluster, (ii) and another set where the headlines come from a different EMM cluster, then we computed the string similarity between those pairs in the same way we did with MSRpar and MSRvid datasets. Accordingly, we sampled 375 headline pairs of headlines that occur in the same EMM cluster, aiming for pairs equally distributed between minimal and maximal similarity using simple string similarity. We sample another 375 pairs from the different EMM cluster in the same manner. A sample pair from the dataset follows:

- Berri says ready to help launch national unity government
- The perception of Spain has changed dramatically with this government

The **SMT** dataset comprises pairs of sentences used in machine translation evaluation. We have two different sets based on the evaluation metric used: an HTER set, and a HYTER set. Both metrics use the TER metric (Snover *et al.* 2006) to measure the similarity of pairs. HTER typically relies on several (1-4) reference translations. HYTER, on the other hand, leverages millions of translations. The HTER set comprises 150 pairs, where one sentence is machine translation output and the corresponding sentence is a human post-edited translation. We sampled the data from the dataset used in the DARPA GALE project with an HTER score ranging from 0 to 120. The HYTER set has 600 pairs from 3 subsets (each subset contains 200 pairs): a. reference vs. machine translation. b. reference vs. Finite State Transducer (FST) generated translation (Dreyer and Marcu 2012). c. machine translation vs. FST generated translation. A sample pair from the dataset follows:

- from calling for the revival of the coptic period to appealing for the reappearance of the ( pharaonic ) era . i just hope that things don 't get any worse.
- the calling to revitalize the coptic epoch is just as bad as the advocacy to revitalize ( the ancient egyptian ) epoch ...

The OnWN/FnWN dataset contains gloss pairs from two sources: OntoNotes-WordNet (**OnWN**) and FrameNet-WordNet (**FnWN**). These pairs are sampled based on the string similarity ranging from 0.4 to 0.9. String similarity is used to measure the similarity between a pair of glosses. The OnWN subset comprises 561 gloss pairs from OntoNotes 4.0 (Hovy *et al.* 2006) and WordNet 3.0 (Christiane Fellbaum 1998). 370 out of the 561 pairs are sampled from the 110K sense-mapped pairs as made available from the authors. The rest, 291 pairs, are sampled from unmapped sense pairs with a string similarity ranging from 0.5 to 0.9. The FnWN subset has 189 manually mapped pairs of senses from FrameNet 1.5 (Baker *et al.* 1998) to WordNet 3.1. They are randomly selected from 426 mapped pairs. In combination, both datasets comprise 750 pairs of glosses. A sample pair from **OnWN** follows:

- measure the depth of a body of water
- any large deep body of water.

And another sample from **FnWN**:

- a certain idiosyncrasy belongs to an entity distinguishing it from other entities.
- unique or specific to a person or thing or category;

### 3.3.3 STS 2014 datasets

The 2014 dataset (Agirre *et al.* 2014) comprises pairs of news headlines (**HDL**), pairs of glosses (**OnWN**), image descriptions (**Images**), DEFT-related discussion forums (**Deft-forum**) and news (**Deft-news**), and tweet comments and newswire headline mappings (**Tweets**).

For **HDL**, we repeated the same process we used the previous year, but avoiding the temporal overlap of the news. For **OnWN**, we used the sense definition pairs of OntoNotes (Hovy *et al.* 2006) and WordNet (Fellbaum 1998). The differences from the previous task is that the two definition sentences in a pair belong to different senses, and that we sampled the pairs based on a string similarity ranging from 0.5 to 1 instead from 0.4 to 0.9.

Inspired by the MSRvid dataset we included a new dataset this year. The **Images** dataset is a subset of the PASCAL VOC-2008 dataset (Rashtchian *et al.* 2010), which consists of 1,000 images with around 10 descriptions each. Just as in MSRvid, the authors asked annotators to provide a one-sentence description of the picture. It was sampled from string similarity values between 0.6 and 1. We

### 3.3. GATHERING DATASETS FOR STS

---

organized two bins with 375 pairs each: one with pairs of descriptions from the same image, and the other one with pairs of descriptions from different images. A sample pair from the dataset follows:

- Two red buses parked up with gardens in front of them.
- Red buses are parked by a large white building beside a formal garden.

**Deft-forum** and **Deft-news** dataset are derived from DEFT data<sup>2</sup>. **Deft-forum** contains the forum post sentences, and **Deft-news** are news summaries. We selected 450 pairs for Deft-forum and 300 pairs for Deft-news. They are sampled evenly from string similarities falling in the interval 0.6 to 1. A sample pair from **Deft-forum** follows:

- The whole earth combined produces enough food for billion people.
- Who does it now only produce enough for billion?

And another sample from **Deft-news**:

- safe bourada was sentenced to 15 years in prison.
- djamel badaoui was sentenced to five years.

The **Tweets** dataset contains tweet-news pairs selected from the corpus released in (Guo *et al.* 2013), where each pair contains a sentence that pertains to the news title, while the other one represents a Twitter comment on that particular news. They are evenly sampled from string similarity values between 0.5 and 1. A sample pair from the dataset follows:

- Broken limbs, torn lives in northern #Mali #Africa #conflict #humanrights
- Broken limbs, torn lives in Mali

---

<sup>2</sup>LDC2013E19, LDC2012E54

### 3.3.4 STS 2015 datasets

The 2015 dataset (Agirre *et al.* 2015a) comprises pairs of sentences from news headlines (**HDL**), image descriptions (**Images**), answer pairs from a tutorial dialogue system (**Answers-student**), answer pairs from Q&A websites (**Answers-forum**), and pairs from a committed belief dataset (**Belief**).

For **HDL** and **Images** we repeated the same process we used the previous year, but again avoiding the temporal overlap of the news and discarding those description pairs that had been already used, respectively. Accordingly, this year we sampled 1000 headline pairs of headlines (instead of just 375, see explanation in Section 3.4.1) that occur in the same EMM cluster, aiming for pairs equally distributed between minimal and maximal similarity using simple string similarity as a metric. We sampled another 1000 pairs from the different EMM cluster in the same manner. Similarly, we organized two bins with 1000 image descriptions pairs each: one with pairs of descriptions from the same image, and the other one with pairs of descriptions from different images.

The source of the **Answers-student** pairs is the *BEETLE corpus* (Dzikovska *et al.* 2010), which is a question-answer dataset collected and annotated during the evaluation of the BEETLE II tutorial dialogue system. The BEETLE II system is an intelligent tutoring engine that teaches students basic electricity and electronics. The corpus was used in the student response analysis task of Semeval-2013. Given a question, a known correct '*reference answer*' and the '*student answer*', the goal of the task was to assess whether student answers were correct, contradictory or incorrect (partially correct, irrelevant or not in the domain). For STS, we selected pairs of answers made up of single sentences. We sampled 2000 pairs using string similarity values between 0.6 and 1. A sample pair from the dataset follows:

- the terminal is separated from the battery terminal
- the terminals are separated by a gap

The **Answers-forums** dataset consists of paired answers collected from the Stack Exchange question and answer websites<sup>3</sup>. Some of the paired answers are responses to the same question, while others are responses to different questions. Each answer in the pair consists of a statement composed of a single sentence or sentence fragment. For multi-sentence answers, we extracted the single sentence from the larger answer that appears to best summarize the answer. We sampled

---

<sup>3</sup><http://stackexchange.com/>

2000 pairs using string similarity values between 0.6 and 1. A sample pair from the dataset follows:

- I don't think there are likely to be any standards that address this issue specifically.
- You're going to find answers all over the map for this one (i.e., there probably aren't "standards").

The **Belief** pairs were collected from the DEFT Committed Belief Annotation dataset (LDC2014E55). All source documents are from English Discussion Forum. We sampled 2000 pairs using string similarity values between 0.5 and 1. It is worth noting that the similarity values were skewed, with very few pairs above 0.8 similarity. A sample pair from the dataset follows:

- stick the cretins in with the people who have a high capacity for learning!
- this is how the people on the right here treat truth.

#### 3.3.5 STS 2016 datasets

The 2016 dataset ([Agirre et al. 2016a](#)) comprises pairs of sentences from news headlines (**HDL**), short answers to computer science questions (**Plagiarism**), post-edited machine translated sentences (**Postediting**), and question-question/answer-answer pairs from *Stack Exchange Data Dump* (**Question-Question & Answer-Answer**). Again, the selection procedure for HDL was the same we used in 2015. For the other datasets pairs are heuristically selected using a combination of lexical surface form and word embedding similarity between a candidate pair of text snippets (see below).

The **Plagiarism** dataset is based on ([Clough and Stevenson 2011](#))'s *Corpus of Plagiarised Short Answers*. This corpus provides a collection of short answers to computer science questions that exhibit varying degrees of plagiarism from related Wikipedia articles. The short answers include text that was constructed by each of the following four strategies:

1. Copying and pasting individual sentences from Wikipedia.
2. Light revision of material copied from Wikipedia.
3. Heavy revision of material from Wikipedia.
4. Non-plagiarised answers produced without even looking at Wikipedia.

## CHAPTER 3. SEMANTIC TEXTUAL SIMILARITY

---

This corpus was segmented into individual sentences using CoreNLP (Manning *et al.* 2014). A sample pair from the dataset follows:

- $P(B)$  is the prior or marginal probability of  $B$ , and acts as a normalizing constant.
- $P(A)$ , or the probability that the student is a girl regardless of any other information.

For **Postediting** we used the (Specia 2011) *EAMT 2011 corpus*, which provides machine translations of French news data using the Moses machine translation system (Koehn *et al.* 2007) paired with postedited corrections of those translations. The corrections were provided by human translators instructed to perform the minimum number of changes necessary to produce a publishable translation. STS pairs for this evaluation set are selected both using the surface form and embedding space pairing heuristics and by including the existing explicit pairs of each machine translation with its postedited correction. A sample pair from the dataset follows:

- This is what we think is the most contributions from the point of view of customers.
- I believe that in the photograph the portrait is the kind of the most difficult.

The **question-question** and **answer-answer** evaluation sets are extracted from the Stack Exchange Data Dump (Stack Exchange, Inc. 2016). The data include long form Question-Answer pairs on a diverse set of topics ranging from highly technical areas such as programming, physics and mathematics to more casual topics like cooking and travel. Pairs are constructed using questions and answers from the following less technical Stack Exchange sites: *academia, cooking, coffee, diy, english, fitness, health, history, lifehacks, linguistics, money, movies, music, outdoors, parenting, pets, politics, productivity, sports, travel, workplace* and *writers*. Since both the questions and answers are long form, often being a paragraph in length or longer, heuristics are used to select a one sentence summary of each question and answer. For questions, we use the title of the question when it ends in a question mark (questions with titles not ending in a '?' are discarded). For answers, a one sentence summary of each question is constructed using LexRank (Erkan and Radev 2004) as implemented by the Sumy<sup>4</sup> package. A sample pair from **question-question** follows:

- Should I drink water during my workout?

---

<sup>4</sup><https://pypi.python.org/pypi/sumy>



- How can I get my toddler to drink more water?

And another sample from **answer-answer**:

- If you are not sure how to do it, don't do it at all.
- If they don't, don't force it.

### Heuristics for pairs selection

These heuristics are used to find pairs sharing some minimal level of either surface or embedding space similarity. An approximately equal number of candidate sentence pairs are produced using our lexical surface form and word embedding selection heuristics. Both heuristics make use of a *Penn Treebank* style tokenization of the text provided by CoreNLP (Manning *et al.* 2014).

**Surface Lexical Similarity** is a surface form selection heuristic that uses an information theoretic measure based on unigram overlap (Lin 1998b). As shown in equation (3.1), surface level lexical similarity between two snippets  $s_1$  and  $s_2$  is computed as a log probability weighted sum of the words common to both snippets divided by a log probability weighted sum of all the words in the two snippets.

$$\text{sim}_l(s_1, s_2) = \frac{2 \times \sum_{w \in s_1 \cap s_2} \log P(w)}{\sum_{w \in s_1} \log P(w) + \sum_{w \in s_2} \log P(w)} \quad (3.1)$$

Unigram probabilities are estimated over the evaluation set data sources and are computed without any smoothing.

**Word Embedding Similarity** is the second heuristic, which computes the cosine between a simple embedding space representation of the two text snippets. Equation (3.2) illustrates the construction of the snippet embedding space representation,  $\mathbf{v}(s)$ , as the sum of the embeddings for the individual words,  $\mathbf{v}(w)$ , in the snippet. The cosine similarity can then be computed as in equation (3.3).

$$\mathbf{v}(s) = \sum_{w \in s} \mathbf{v}(w) \quad (3.2)$$

$$\text{sim}_v(s_1, s_2) = \frac{\mathbf{v}(s_1) \cdot \mathbf{v}(s_2)}{\|\mathbf{v}(s_1)\| \|\mathbf{v}(s_2)\|} \quad (3.3)$$

Three hundred dimensional word embeddings are obtained by running the *GloVe* package (Pennington *et al.* 2014) with default parameters over all the data

collected from the 2016 evaluation sources.<sup>5</sup>

## 3.4 Annotation

The next step after gathering the sentences for each dataset is to get quality assessments. The process of annotating the datasets for STS has been changing over the years, as we have been learning and gaining experience. Despite this, the process is mostly the same each year, with small differences. The starting point is as follows:

1. We have defined a straightforward Likert scale ranging from 5 to 0, and we decided to provide definitions for each value in the scale (cf. Figure 3.1).
2. We have gathered sentences from several sources and created different datasets.

Annotating these datasets involves time and money, as it does when you want to build any knowledge base. Therefore, before spending a considerable amount of money in a large-scale annotation process (for the 2012 datasets) we did some pilot annotations. We selected 200 pairs at random from the three main datasets in the training set (**MSRpar**, **MSRvid** and **SMT-Europarl**). We personally did these annotation, and the pairwise Pearson correlation ranged from 84% to 87% among ourselves. The agreement of each annotator with the average scores of the other was between 87% and 89%. This was preliminary evidence that the task was well defined.

Given the good results of the pilot we decided to deploy the task in AMT in order to crowd source the annotation task. As mentioned above, the annotation process has evolved over the years, although these differences are small. The biggest difference is that in 2013 we did not use AMT, but *CrowdFlower*, but this was extraordinary, as we returned to AMT the following years. Annotators were presented with the detailed instructions provided in Figure 3.1, and were asked to label each STS sentence pair between 0 and 5, selecting from a dropdown box. Five sentence pairs were presented to each annotator at once, per *Human Intelligence Task* (HIT), at a payrate of \$0.20. We collected five separate annotations

---

<sup>5</sup>The evaluation source data contained only 10,352,554 tokens. This is small relative to the datasets used to train embedding space models that typically make use of > 1B tokens. However, the resulting embeddings are found to be functionally useful for finding semantically similar text snippets that differ in surface form.

per sentence pair. The turkers were required to have achieved a 95% of approval rating in their previous HITs, and had to pass a qualification task which included 6 example pairs. Annotators were restricted to people from the following countries: Australia, Canada, India, New Zealand, UK, and US. From 2014 onwards, annotators were only eligible to work on the task if they had the *Mechanical Turk Master Qualification*, a special qualification conferred by AMT (using a priority statistical model) to annotators who consistently maintain a very high level of quality across a variety of tasks from numerous requesters. Access to these skilled workers entails a 20% surcharge.

To **monitor the annotations** of the crowdsourcing workers, we created a representative gold dataset of 105 pairs that were manually annotated by the task organizers during STS 2013. We include one of these gold pairs in each set of five sentence pairs, where the gold pairs are indistinguishable from the rest. Unlike when we ran on CrowdFlower for STS 2013, these gold pairs are not used for training purposes, nor are workers automatically banned from the task if they make too many mistakes on annotating them. Rather, the gold pairs are only used to help in identifying and removing the data associated with poorly performing annotators. With few exceptions, 90% of the answers from each individual annotator fall within +/-1 of the answers selected by the organizers for the gold dataset.

In the next section we detail a **post-hoc validation** process performed to improve the quality of the datasets.

### 3.4.1 Quality of annotation

In the previous section we have seen how we used *Gold Standard* (GS) pairs to evaluate the annotations. These GS pairs were used to discard the annotations of low quality workers. This quality control is done during the annotation process. In addition to this **pre-annotation** quality control, we perform an additional **post-annotation** quality control, which we describe in this section.

In order to assess the annotation quality, we measured the correlation of each annotator with the average of the rest of the annotators. We can see this average score of the rest of the annotators as a GS. If any annotator gets a very low correlation score (below 0.5) in respect to this virtual GS we remove this annotator and all his/her annotations. We then averaged all the other annotations to create the final dataset. This method to estimate the quality is identical to the method used for evaluation (see Section 3.5.1) and it can be thus used as the upper bound for the systems. An example of this filtering is shown below:

## CHAPTER 3. SEMANTIC TEXTUAL SIMILARITY

---

turkers with average of others:

GS:10014836.average SYS:10014836 N:5 Pearson: 0.969

GS:09100711.average SYS:09100711 N:5 Pearson: 0.935

GS:09843876.average SYS:09843876 N:5 Pearson: 0.966

GS:09872120.average SYS:09872120 N:5 Pearson: 0.910

GS:15460725.average SYS:15460725 N:5 Pearson: 0.475

In the example above each line represents an annotator. The first column is the average score of the other annotators that evaluated the same pair, the second column is the ID of the annotator, the third column is the total number of pairs that annotator evaluated, and the last column is the correlation of the annotator in respect to the virtual GS. Take into account that the four other annotators with we computed the GS may be different for each of the five pair of sentences.

In another attempt to improve the quality of the data, in 2015 we used an additional filtering process. We selected 2000 pairs (instead of 750) from each dataset and annotated all of them. This 'raw' data was automatically filtered in order to achieve the following three (partially conflicting) goals:

1. Obtain a more uniform distribution across scores.
2. Select pairs with high inter-annotator agreement.
3. Select pairs which were difficult for a string-matching baseline.

This filtering process was purely automated and involved no manual selection of pairs. The raw annotations and the Perl scripts that generated the final gold standard are available at the task website.

The final number of selected pairs per dataset is shown in Table 3.1, and the inter-tagger correlation for each dataset is shown in Table 3.2. The correlation figures are generally very high (over 70%). The post-filtering process helps to increase the inter-tagger correlation. The datasets with lower values are **SMT-Europarl**, **SMT-News**, **Deft-forum**, and **OnWN** (2012). This happens in datasets that are generally more difficult to understand, because they are not written in a natural way. We have seen examples of each of the datasets in Section 3.3, but we show again an example from **OnWN** (2012):

- seal, insulate or protect
- treat the body or any part of it by wrapping it, as with blankets or sheets, and applying compresses to it, or stuffing it to provide cover, containment, or therapy, or to absorb blood.

Year	Dataset	Inter-tagger correlation
2012	MSRpar	71.2%
	MSRvid	87.4%
	OnWN	62.9%
	SMT-Europarl	53.0%
	SMT-News	56.4%
2013	FNWN	69.9%
	HDL	85.0%
	OnWN	87.2%
	SMT	65.8%
2014	Deft-forum	58.6%
	Deft-news	70.7%
	HDL	79.4%
	Images	83.6%
	OnWN	67.2%
	Tweets-news	74.4%
2015	Answer-forums	64.7%
	Answer-students	76.6%
	Belief	73.8%
	Headlines	82.1%
	Images	84.6%
2015 (Post-filtered)	Answer-forums	74.2%
	Answer-students	82.2%
	Belief	72.1%
	Headlines	86.9%
	Images	88.8%

**Table 3.2** – Inter-tagger correlation scores for each of the datasets. The difference between the correlation of OnWN 2012 and 2013 datasets may be because in 2013 CrowdFlower was used for the annotation process, rather than AMT.

And another example of unnaturally written sentences, from **Deft-forum** (2014):

- Jake with the assist.
- Voracek with the goal.

The distribution of scores is not the same among the different dataset. The **HDL** dataset tends to be uniform, but the scores for **SMT** are not uniform, with

most of the scores uniformly distributed between 3.5 and 5, a few pairs between 2 and 3.5, and nearly no pairs with values below 2. The source data for each of the datasets has a big relevance in the distribution of the scores. For instance, if we gathered sentences from a source with a fixed topic such as the **Answers-student** we are going to obtain less values below 2 because all the sentences share the topic. The datasets with the highest average scores are those derived from SMT sources, with average scores between 4.55 and 4.19. On the other hand, the datasets with the lowest average scores are the **Answer-Forums**, **Belief** and **FNWN**, being their average score lower than 2. The datasets with the best distribution are **HDL13**, **HDL15**, **HDL16**, **Postediting**, **Answer-Answer**, **Plagiarism** and **Images15**, which average score is around 2.5.

The distribution of scores for different datasets is shown in Figure 3.3. Additionally, the minimum, maximum, average and standard deviation values of all the dataset are shown in Table 3.3.

## 3.5 System Evaluation

The objective of creating the datasets is to allow the participants to create systems for STS and to evaluate them with respect to the systems of the other participants. These systems should return a similarity score for any two sentences. Deciding how to evaluate the output of these systems and ranking them was not easy.

In the next section we are going to explain the evaluation metrics we used for STS, and how they evolved over the years. Next in Section 3.5.2 we present the baseline systems used in the competitions. In Section 3.5.3 we briefly discuss the participation the task attracted. Finally, in Section 3.5.4 we present the best systems of each of the year and the tools and resources they employed in Section 3.5.5.

### 3.5.1 Evaluation metrics

In 2012, evaluation of STS was still an open issue. In order to have a single Pearson measure for each system we concatenated the gold standards (and system outputs) for all 5 datasets into a single gold standard file (and a single system output). The first version of the results were published using this method, but the overall score did not correspond well to the individual scores in the datasets, and participants proposed two additional evaluation metrics, both of them based on Pearson correlation. We decided that it was more informative, and on the benefit of the community, to also adopt those evaluation metrics, and the idea of having a

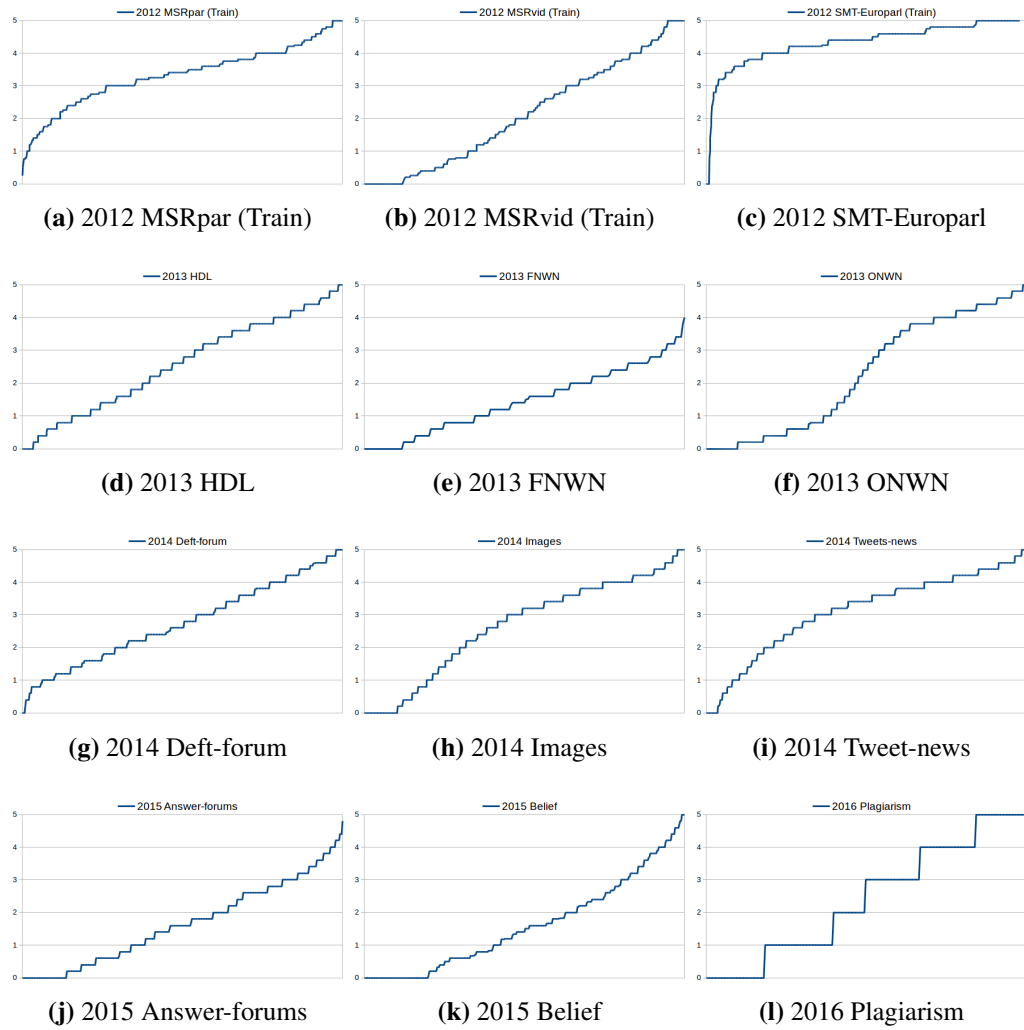
### 3.5. SYSTEM EVALUATION

Year	Dataset	Min	Max	Avg	Sdv
2012	MSRpar (Train)	0.25	5.00	3.32	0.93
	MSRvid (Train)	0.00	5.00	2.14	1.60
	SMT-Europarl (Train)	0.00	5.00	4.31	0.71
	MSRpar (Test)	0.75	5.00	3.27	0.92
	MSRvid (Test)	0.00	5.00	2.30	1.64
	SMT-Europarl (Test)	1.50	5.00	4.55	0.52
	SMT-News	0.25	5.00	4.33	0.82
	OnWN12	0.00	5.00	3.87	1.02
2013	FnWN	0.00	4.00	1.47	0.99
	HDL13	0.00	5.00	2.57	1.42
	OnWN13	0.00	5.00	2.31	1.76
	SMT	1.20	5.00	4.19	0.57
2014	Deft-forum	0.00	5.00	2.75	1.25
	Deft-news	0.00	5.00	3.03	1.26
	HDL14	0.00	5.00	2.77	1.40
	Images14	0.00	5.00	2.67	1.50
	OnWn14	0.00	5.00	2.64	1.93
	Tweets-news	0.00	5.00	3.11	1.27
2015	Answer-forums	0.00	4.80	1.66	1.23
	Answer-students	0.00	5.00	2.92	1.41
	Belief	0.00	5.00	1.62	1.38
	HDL15	0.00	5.00	2.56	1.62
	Images15	0.00	5.00	2.50	1.65
2016	Answer-Answer	0.00	5.00	2.49	1.75
	HDL16	0.00	5.00	2.47	1.71
	Plagiarism	0.00	5.00	2.42	1.76
	Postediting	0.00	5.00	2.51	1.73
	Question-Question	0.00	5.00	2.13	1.50

**Table 3.3** – Minimum, maximum, average and standard deviation values for each dataset. Standard deviation values are quite low for SMT based datasets: SMT-Europarl (Train), SMT-News, and SMT.

single main evaluation metric was dropped. This decision was not taken without controversy, but the organizers gave more priority to openness and inclusiveness and to the involvement of participants. The final result table thus included three

## CHAPTER 3. SEMANTIC TEXTUAL SIMILARITY



**Figure 3.3** – Average scores for each pair in datasets between 2012 and 2016. In the x axis the pairs ordered according to mean score in increasing order. In the y axis the mean score for each pair.

evaluation metrics.

The first evaluation metric was the Pearson correlation for the concatenation of all five datasets, as described above. We used *overall Pearson* or simply *ALL* to refer to this measure.

The second evaluation metric normalizes the output for each dataset sepa-



rately, using the *linear least squares* method. We concatenated the system results for five datasets and then computed a single Pearson correlation. Given  $Y = \{y_i\}$  and  $X = \{x_i\}$  (the gold standard scores and the system scores, respectively), we transform the system scores into  $X' = \{x'_i\}$  in order to minimize the squared error  $\sum_i (y_i - x'_i)^2$ . The linear transformation is given by  $x'_i = x_i * \beta_1 + \beta_2$ , where  $\beta_1$  and  $\beta_2$  are found analytically. We refer to this measure as *Normalized Pearson* or simply *ALLnorm*. This metric was suggested by one of the participants, Sergio Jimenez.

The third evaluation metric was the weighted mean of the Pearson correlations on individual datasets. The Pearson returned for each dataset is weighted according to the number of sentence pairs in that dataset. Given  $r_i$  the five Pearson scores for each dataset, and  $n_i$  the number of pairs in each dataset, the weighted mean is given as  $\sum_{i=1..5} (r_i * n_i) / \sum_{i=1..5} n_i$ . We refer to this measure as *weighted mean of Pearson* or *Mean* for short.

From this year on, we used the weighted mean of the Pearson correlations to aggregate the results from each dataset into an overall score. The analysis performed in (Agirre and Amigó In prep.) shows that Pearson and averaging across datasets are the best suited combination in general. In particular, Pearson is more informative than Spearman, in that Spearman only takes the rank differences into account, while Pearson does account for value differences as well.

### 3.5.2 The baseline system

The scores were produced using a simple word overlap baseline system named **TokenCos**. The input sentences were tokenized splitting at white spaces, and then represented each sentence as a vector in the multidimensional token space. Each dimension had 1 if the token was present in the sentence, 0 otherwise. Similarity of vectors was computed using cosine similarity. This baseline ranked 77/88, 73/89, 27/38, 61/74 and 100/113 in 2012, 2013, 2014, 2015 and 2016, respectively. The drop in performance between the best system and the TokenCos baseline in each of the years was the following: -0.24, -0.25, -0.25, -0.21 and -0.26. The difference between the baseline and the best systems is quite stable over the different competitions. One explanation of this is that, despite the improvements of the systems, the datasets are also more difficult each year.

In 2013 we also run two freely available systems, **DKPro** (Bar et al. 2012) and **TakeLab** (Šarić et al. 2012) from STS 2012. They served as two strong contenders since they ranked 1st (DKPro) and 2nd (TakeLab) in 2012 year's STS task. We trained the *DKPro* and *TakeLab-sts12* models on all the training and test

data from STS 2012 and evaluated them on the 2013 datasets. We additionally trained another variant system of TakeLab, *TakeLab-best*, where we used targeted training where the model yielded the best performance for each test subset as follows: (1) HDL was trained on MSRpar 2012 data; (2) OnWN was trained on all 2012 data; (3) FnWN was trained on 2012 OnWN data; (4) SMT was trained on 2012 SMTeuroparl data. Note that Takelab-best is an upper bound, as the best combination is selected on the test dataset. TokenCos, TakeLab-sts12, TakeLab-best, DKPro would rank as 70th, 58th, 27th and 6th among 89 system submissions, respectively. Takelab-best's result was less than 10 points below the best system, while Takelab-sts12's result was more than 18 below. The different results yielded from TakeLab depending on the training data suggests that some STS systems are quite sensitive to the source of the sentence pairs, indicating that domain adaptation techniques is important for this task. DKPro performed extremely well when trained on all available training, only 5 points below the best system, with no special tweaking for each dataset.

TakeLab was also used as a competitive baseline in 2014 and 2015, and was trained with all datasets from previous years. TakeLab would rank 18th in 2014 and 42nd in 2015, ten absolute points below the best system in both cases, a bigger difference than in 2013. This shows that systems are getting better.

### 3.5.3 Participation

To participate in the task teams are required to register in advance. Participants should download the test sets, run their systems on it and upload the output of their systems in text format. The first year the train datasets were released two months before the test data, and from there on the datasets of previous years were used as train. The test datasets are released on the previously announced date.

The evaluation windows is typically of 15 days, but after downloading the test datasets they have a maximum of 120 hours to upload the results. Participants could send a maximum of three system runs. After the submission deadline expired, the organizers published the gold standard in the task website, in order to ensure a transparent evaluation process. The number of participants and the number of systems runs sent each year are listed in the table 3.4.

The task has shown to attract great interest, always being the first or second task of SemEval by number of participants. The decrease shown for 2014 was a consequence of announcing the task later than usual.

Year	Teams	Runs
2012	35	88
2013	34	89
2014	15	38
2015	29	74
2016	43	119

**Table 3.4** – Participants in the STS task by year.

### 3.5.4 Results

This section describes the best STS systems that participated in the competition. First of all, it is important to stress that the large majority of the systems are well above the simple baseline. Table 3.5 shows the results for the baselines and for the best three runs of each year. Each result is ordered by the rank of the system according to the weighted mean of the Pearson correlations on individual datasets. In addition, the Pearson score for each dataset is given.

The first thing that comes to mind from the results of Table 3.5 is that the mean correlation of the systems has been rising steadily over the years up to the year 2016. It seems that the method for selecting more difficult pairs used that year (see Section 3.3.5) works well. The difference between the best systems is very small, in many cases not being statistically significant.

As for the individual datasets, the highest correlations are obtained for MSRvid 2012 (0.880), OnWN 2014 (0.8745), Images 2015 (0.864) and Postediting 2016 (0.848), and the lowest for SMT 2013 (0.327), SMT-news 2012 (0.399), Deft-forums 2014 (0.471) and SMT-Europarl 2012 (0.477). Datasets with higher correlations have in common the simplicity of their sentences, which are mostly quite short. On the contrary, the datasets with lower correlations are usually datasets with long cryptic sentences, that are hard to understand. In general, the correlation for the non-MT datasets is really high.

Another aspect to keep in mind is that some systems are very dependent on training data. For instance, the *NTNU-run3* system (2014) obtains the highest correlation in 4 datasets out of 6, but was ranked 3rd because it performed poorly on the Tweet-news dataset: 9 points below the best system on overall, and more than 10 points below the best system on that dataset. However, this loss of performance could also be explained by an incorrect/missing sentence pre-processing.

CHAPTER 3. SEMANTIC TEXTUAL SIMILARITY

Team and run	MSRpar	MSRvid	SMT- eur	OnWN 12	SMT- news		Mean
UKP-run2	0.683	0.874	<b>0.528</b>	0.664	<b>0.494</b>		<b>0.677</b>
TakeLab-simple	<b>0.734</b>	<b>0.880</b>	0.477	0.680	0.399		0.675
Soft-Cardinality	0.640	0.856	0.515	<b>0.711</b>	0.483		0.671
baseline	0.433	0.300	0.454	0.586	0.391		0.436
Team and run	HDL 13	OnWN 13	FNWN	SMT			Mean
UMBC-run1	<b>0.764</b>	0.753	<b>0.582</b>	<b>0.380</b>			<b>0.618</b>
UMBC-run2	0.743	0.705	0.544	0.371			0.593
Deft-base	0.653	<b>0.843</b>	0.508	0.327			0.580
DKpro	0.735	0.735	0.341	0.326			0.565
TakeLab-best	0.656	0.633	0.405	0.339			0.522
Takelab-sts12	0.486	0.633	0.269	0.279			0.434
baseline	0.540	0.283	0.215	0.286			0.364
Team and run	Deft- forum	Deft- news	HDL 14	Images 14	OnWN 14	Tweet- news	Mean
DLS@CU-run2	0.483	0.766	0.765	0.821	<b>0.859</b>	0.764	<b>0.761</b>
Meerkat	0.471	0.763	0.760	0.801	0.875	<b>0.779</b>	0.761
NTNU-run3	<b>0.531</b>	<b>0.781</b>	<b>0.784</b>	<b>0.834</b>	0.850	0.676	0.755
TakeLab	0.333	0.716	0.720	0.742	0.793	0.650	0.678
baseline	0.353	0.596	0.510	0.513	0.406	0.654	0.507
Team and run	Ans- forum	Ans- stu.	Belief	HDL 15	Images 15		Mean
DLS@CU-S1	<b>0.739</b>	0.773	<b>0.749</b>	<b>0.825</b>	<b>0.864</b>		<b>0.802</b>
ExBThemis	0.695	<b>0.778</b>	0.748	<b>0.825</b>	0.853		0.794
DLS@CU-S2	0.724	0.757	0.722	<b>0.825</b>	0.863		0.792
baseline	0.445	0.665	0.652	0.531	0.604		0.587
Team and run	Ans.- Ans.	HDL 16	Plag.	Post- edit	Ques.- Ques.		Mean
Samsung-EN1	<b>0.692</b>	<b>0.827</b>	<b>0.841</b>	0.835	0.687		<b>0.778</b>
UWB	0.621	0.819	0.824	0.821	0.702		0.757
Mayo-run3	0.614	0.773	0.805	<b>0.848</b>	<b>0.747</b>		0.756
baseline	0.411	0.541	0.696	0.826	0.038		0.513

**Table 3.5** – Best three runs and baselines for each of the years.

### 3.5.5 Tools and Resources

In addition to the system output, organizers asked participants to submit a description file, special emphasis on the tools and resources that they used. Given the number of participants, it is really complicated to summarize all the tools and resources used by the participants. In the first editions of the task, the totals showed that WordNet was the most used resource, followed by monolingual corpora and Wikipedia. Acronyms, dictionaries, multilingual corpora, stopword lists and tables of paraphrases were also used. In the last years, aligning words between sentences has been the most popular approach for the top participants. They use WordNet (Christiane Fellbaum 1998), *Word Embeddings* (Mikolov *et al.* 2013a; Baroni *et al.* 2014a) and PPDB. In general, generic NLP tools such as lemmatization, PoS tagging, distributional word embeddings, distributional and knowledge-based similarity are widely used, and to a lesser extent, parsing, word sense disambiguation, semantic role labelling and time and date resolution. Most teams add a machine learning algorithm to learn the output scores, but some team did not use it in their best run.

One observation we made is that all participants use the resources separately. The most recurrent approach is to use the resources to obtain a STS score and then use these values together with others to feed a *Machine Learning* (ML) system. Most systems also add different characteristics extracted from the sentences to enrich the ML system, but no one has worked in the combination of different resources. Inspired by this, in the next chapter we present a novel system that combines different resources to obtain a STS score.

## 3.6 Conclusions

*Semantic Textual Similarity* captures the notion that some texts are more similar than others, measuring their degree of semantic equivalence. Textual similarity can range from complete unrelatedness to exact semantic equivalence, and a graded similarity intuitively captures the notion of intermediate shades of similarity, as pairs of text may differ from some minor nuanced aspects of meaning, to relatively important semantic differences, to sharing only some details, or to simply being related to the same topic.

This chapter presents the STS task from its conception to the present. The task has been very constant over the years, the only changes being the origin of the datasets. Throughout the chapter we have described how the task was designed, its sources of inspiration, and the most similar tasks as TE and PARA.

Recently, STS is also being used to evaluate semantic representations of sentences. Several teams are competing aiming to generate the best sentence representations or vectors, and they are using the datasets for STS to evaluate the quality of their embeddings. (Wieting *et al.* 2015) generates sentence embeddings using neural networks, but also by simple averaging of the word vectors. (Mu *et al.* 2017) follow the previous work on sentence embedding by averaging, adding a post-processing step to improve the sentence representation. (Arora *et al.* 2017) generates sentence representations averaging word vectors computed on Wikipedia and then modifies them using *Principal Component Analysis* an *Singular value Decomposition*, and demonstrates that this simple approach is still a very strong baseline.

STS differs from TE in as much as it assumes symmetric graded equivalence between the pair of textual snippets, and in the case of TE the equivalence is directional. Additionally, STS also differs from both TE and PARA in that STS wants to incorporate the notion of graded semantic similarity rather than being a binary yes/no decision. When we thought for the first time in creating it, the four main objectives were the following:

1. To set a definition of STS as a graded notion which can be easily communicated to non-expert annotators beyond the likert-scale.
2. To gather a substantial amount of sentence pairs from diverse datasets, and to annotate them with high quality.
3. To explore evaluation measures for STS.
4. To explore the relation of STS to PARA and Machine Translation Evaluation (MTE) exercises.

The first three objectives have been fulfilled during the development of this thesis: the task has been defined, the datasets have been created, and an agreement has been reached on how to evaluate the systems. The fourth objective, however, has not been carried out and remains as work for the future.

As a result, STS has become a very popular task. Participation in the task has been the highest of all *SemEval*/*\*SEM* tasks, with the exception of 2014, in which it was the second task with more participants, and STS datasets are widely used in different areas.

After five editions of the competition the definition of the task is already very consolidated. We are aware that the task has been criticized, arguing that it is not completely well defined. However, observing the inter-tagger correlation scores

we can see that they are, in general, very high. High correlations prove that the task is well defined and that this criticism does not make sense.

Another possible criticism to the task is that the differences between the scores 'are too fuzzy'. We are also aware of this, in fact, in the 2014 edition we presented the sub-task of STS in spanish and the organizers of the task merged the ranks 4 and 3 into one. However, we think that this is not important, because STS presents gradual similarity evaluation. It is not important if a scorer assigns a 4 or a 3, since what is really important is if they are in agreement on how to order them, on which pair of sentences is more similar. The distribution of scores in the datasets and the high inter-tagger agreement shows that the objective of achieving gradual similarity values has been achieved.

Finally, contradiction and its place within STS is also a much discussed topic. Working with this phenomenon is not one of the objectives of STS. There are other tasks, such as *Interpretable STS* (iSTS), that do try to solve the contradiction, but that is not the case of STS. In any case, there are not many cases of contradiction in our datasets.

In the same period we have created and annotated 25 datasets, which make a total of 15436 pairs of sentences. This makes them the largest collection of data for STS. The quality of the annotations has been improved gradually each year, rising from an average inter-tagger of approximately 70% in 2012 to an approximately 83% in 2015. Datasets are widely used for the evaluation of semantic similarity and other related tasks.

In the 2012 competition there was a discussion on the best methods for evaluating STS systems. To the official proposal of the organizers two more methods were added, one of which was later adopted as the official, the weighted mean of correlations. In some editions, confidence scores were allowed along with the STS scores, and they proved to be useful. While it is recognized that Pearson has some problems, such as its erratic behaviour in the first edition when the files were concatenated, it has been established as the standard measure for STS evaluation.

As we have seen, we have defined the task, we have created datasets for it, and we have an agreement on how to evaluate the systems. With these premises, the next obvious step is to make the most of the results of this work by creating a high-level system for STS. In the next chapter we describe a system based on our knowledge of the best current systems. We have seen the most commonly used resources and techniques, and we noticed that none have worked using these resources in combination. We truly believe that it is possible to gain greater benefit from the knowledge within them.





## Cubes for Semantic Textual Similarity

In this section we present our system for *Semantic Textual Similarity* (STS). First we introduce the motivation behind this system. In the next section we describe how we build a cube where we store all the information we have available. Then we explain how we make use of the information previously stored in the cube, and how we produce a similarity score. Finally, we describe how we have designed the evaluation of this system, and its comparison with the state-of-the-art.

### 4.1 Motivation

The best performing systems for STS presented in Chapter 2 make use of linear regression models to combine multiple text similarity measures, such as UKP (Bar *et al.* 2012) and Takelab (Šarić *et al.* 2012). Another popular approach for STS is to use align word between sentences and scoring them with some resource (Sultan *et al.* 2014a). Our hypothesis is that we can obtain better results **combining** word-to-word similarity from different sources at the word level, in contrast to other works where each resource is used independently. Consider the following pair of sample sentences:

- *President Obama warns Russia for taking over Ukraine.*
- *John Kerry admonished Russia for invading Ukraine.*

The sentences above mean almost the same but aligning the words from one sentence to the other is not trivial. If we search for the most similar word for

'Obama' in the second sentence it may be 'Kerry', and the most similar for 'warns' may be 'admonished'. Now suppose that we have a resource that is good with named entities, and aligns 'Obama' with 'Kerry' with a high value. Nothing guarantees that this same resource will be able to properly align 'warns' and 'admonished' and score them with a high similarity. The same happens in the case of a resource that is able to do well with 'warns' and 'admonished', which might not do well with 'Obama' with 'Kerry'. In this hypothetical example we would end up with two similarity scores for these sentences, none of them being completely adequate. Even if we use these values to feed a Machine Learning (ML) system and improve the final result, we would use incomplete information.

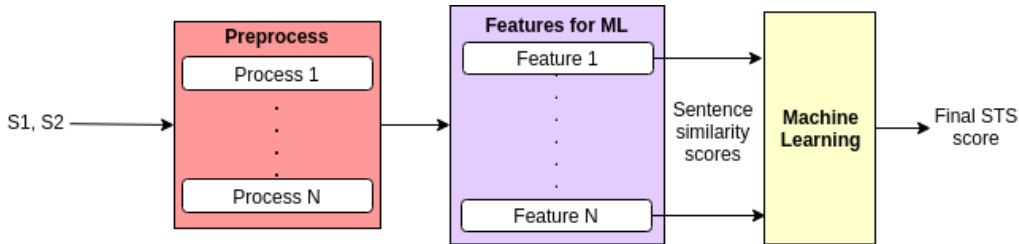
Now consider that we can detect all these alignments using more than one resource at once. The first resource would be able to align 'Obama' with 'Kerry' and the second one would be able to align 'warns' and 'admonished'. But we could also have another resource that would be able to properly align 'taking over' with 'invading' and score it with a high similarity. To achieve this goal it is important to maintain as much knowledge as possible until the last step before the final decision, and if we compute the sentence similarity based on each resource separately we are losing information.

## 4.2 Building Cubes

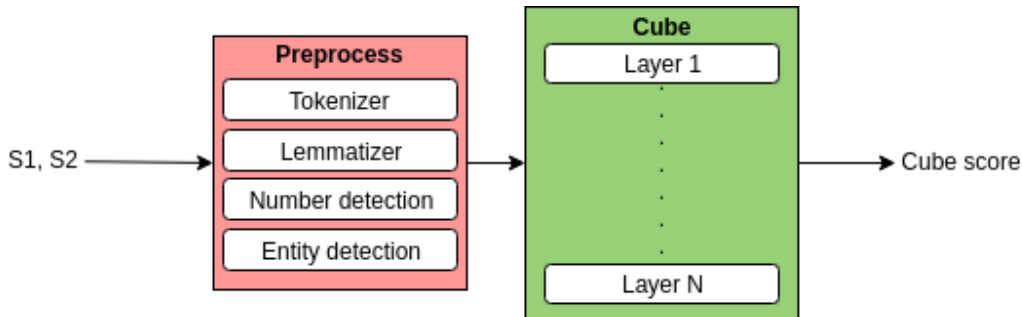
As we have seen in Chapter 2, most systems for STS are based on generating features that are then used to feed an automatic learning system. A diagram of these types of systems can be seen in Figure 4.1. As can be seen, in the intermediate step different features are computed, some of which are already similarity values derived from different resources such as word vectors, WordNet, PPDB or any other resources. That is, these systems generate sentence similarity values (along with other features), which they use to train an STS system.

Our idea is that we can combine these resources in a different way, and not just use them for ML. To do this, we propose the systems presented by the diagrams in Figure 4.2 and Figure 4.3. The former is a system that does not use any ML, taking advantage of the joint knowledge, and extracting a final similarity score which represents the token-wise similarity scores for each resource in different layers. The system in Figure 4.3 uses the output of the former system in combination with other features that are created for ML. Somehow, the second proposal is a combination of our first system and a typical STS system as the one shown in Figure 4.1. It combines features, the cube scores and scores for individual layers. We expect that these two proposals will obtain better results, because we make

better use of the resources than the typical systems shown in Figure 4.1. A more detailed description of these systems is given below.



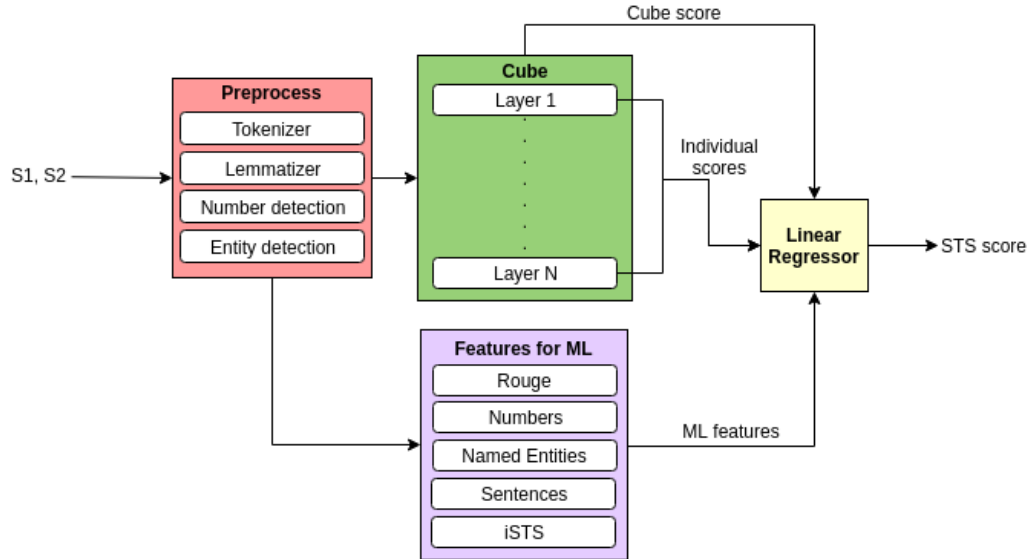
**Figure 4.1** – Flowchart of a typical STS system, where  $feature_i$  corresponds to  $resource_i$ .



**Figure 4.2** – Flowchart of our system without ML, where  $layer_i$  corresponds to  $resource_i$ , and each layer contains token-pair similarities.

In our preprocess step both sentences are tokenized and *lemmatized*, numbers are normalized, and named entities are detected. To tokenize and lemmatize we use the *Stanford parser* (Toutanova *et al.* 2003). When normalizing numbers we search for numbers that are written in letters and convert them into numbers. Their format is also normalized by removing commas (for example, converting 1,000 to 1000). To detect the entities we use the output of the Stanford parser, but we also match them with the entities present in Wikipedia (see Section 4.2.1).

Once we have the words/lemmas from the previous step we can start the construction of the cube. Each pair of sentences can be now represented by a  $N \times M$  matrix, being  $N$  the number of tokens of the first sentence, and  $M$  the number of tokens of the second sentence. We can fill this matrix with pairwise similarity scores from any given source of information. For each of the resources we create



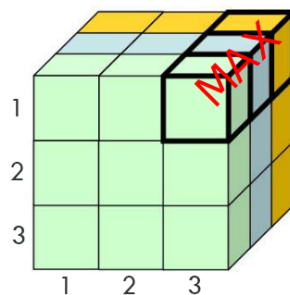
**Figure 4.3** – Flowchart of our system with ML. The STS scores from individual layers, the STS score from the cube and additional features from are used to feed a Linear Regressor.

a layer, where each cell contains the similarity score for a word in the first sentence in respect to a word in the second sentence according to that resource. If we do not find a similarity score for any resource using the words we try using the lemmas in any combination: word/lemma, lemma/word, or lemma/lemma. If we do not find a value for any of the combinations, the corresponding cell will be marked as 'Not-a-number' (NaN) in the layer of that resource. The latter is important because we want to distinguish a zero from a NaN, because it is not the same if a resource gives a 0 ('they are not similar at all') or NaN ('I have no information about that'). For example, the cell (1, 3) of a layer in the cube contains the similarity score of the first word of the first sentence in respect to the third word of the second sentence according to a given resource  $d \in D$  (from now on,  $sim(d, S1_1, S2_3)$ , being  $D$  our layer collection). We can have as many matrices as desired, where each of these matrices is a layer, forming a cube where each dimension reflects a different source of information. For each of these layers, if the tokens/lemmas are the same in both sentences, we assign the highest score to this cell.

We have constructed a cube where we gather similarity values for each word pair combination from the two sentences, according to different sources, and now

we need to select the best score for each pair of words/lemmas. Our hypothesis and our motivation to build this cube is that we can improve the results using token-wise similarities from several sources and combining them, instead of using sentence-wise similarities independently.

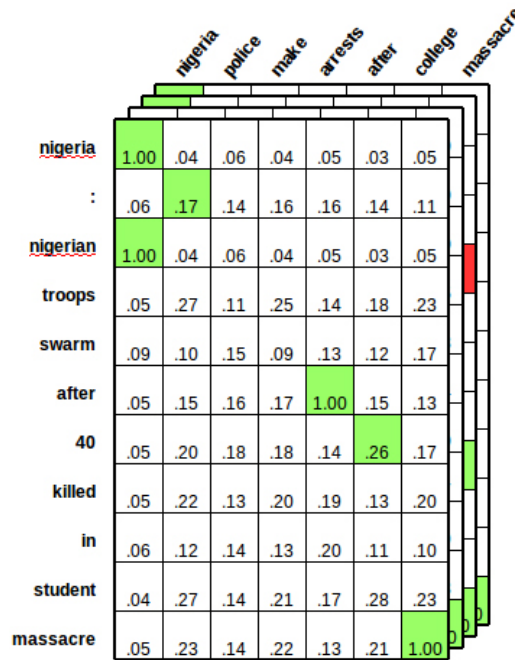
This hypothesis is based on an analysis of examples from Train and Development data using different resources. A **False Negative** (FN) occurs when a resource gives a low value, but it should be high. This could indicate that the resource does not have the necessary information to give a high value. We say **False Positive** (FP) is less frequent, since it happens when a resource assigns a high value, but it should be low. There are times when it may occur due to some phenomena such as polysemy, but these cases are not very common according to what we found in our analysis. This analysis showed that the resources have far more FN than FP. Based on this observation, if one of the resources says that two words are very similar then we trust it, and take that similarity score even if the other resources yield a very low similarity. In the previous section we showed an example of how selecting the highest similarity scores among different resources can help in the task of evaluating two sentences. Thus, following this hypothesis we are going to select the highest score for each word pair in the cube. In other words, we align each word in the first sentence to the word in the second sentence with which it has the highest score among all layers (and vice versa). An abstract visualization of a cube formed by two sentences of length three and with three layers is shown in Figure 4.4 (each layer is represented with a different color). In this example The '*MAX*' in the figure represents that we are going to select the



**Figure 4.4** – Abstract representation of a  $3 \times 3$  cube with three layers (each color is a layer) where we are selecting the maximum value (among all layers) in the top right corner. This cell contains three similarity scores (one per resource) for the 1st word in the first sentence with respect to the 3rd word in the second sentence.

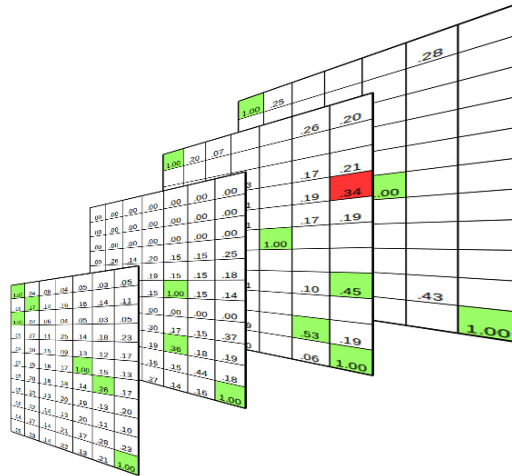
highest value in any layer  $d$  as the similarity for  $sim(d, S1_1, S2_3)$ .

A more complex example of a  $11 \times 7$  cube with four layers is shown in Figure 4.5. **Good** alignments are marked in green, and **bad** alignments are marked in red. In this particular case 'troops' has been aligned with 'massacre' with a score of 0.34, assigned by the third layer, when it should be aligned to 'police'. In some cases it is not possible to make 1-1 alignments, either because the sentences are of different lengths, or because there is no correspondence. In those cases, even if the alignment seems to be incorrect, we consider them as good alignments if the score is low (they are aligned, but they don't have much influence). In this example we aligned ':' with 'police', or '40' to 'college', but scores are low. We can see the cube unfolded in Figure 4.6. As we can see, the first two layers are completely filled, while the third and the fourth are more sparse (see Section 4.2.1 for further information).



**Figure 4.5** – Representation of a  $11 \times 7$  cube with four layers. Good alignment are marked in green, and bad alignments in red.

Once we have the cube we can extract a similarity score using the pairwise similarity score (Sections 4.3.1 and 4.3.2) and the features generated for ML (Section 4.3.4). Before that, the following section describes all the layers with which



**Figure 4.6** – Unfolded representation of a  $11 \times 7$  cube with four layers. The first two layers are dense layers and the other two layers are more sparse. Good alignment are marked in green, and bad alignments in red.

the cube is built.

### 4.2.1 Layers of the cube

This sections describes all the layers that form the cube. In general, we can distinguish two types of layers: distributional and those derived from knowledge bases. The final cube is composed of eight layers, but we tried some other resources in development. First, we describe the eight layers, and then we briefly explain some of the discarded layers and the reason why they were discarded.

Representing words as vectors has become a popular way to address several NLP task such as *Textual Entailment* (Dagan *et al.* 2010), *Paraphrase Detection* (Dolan and Brockett 2005), *Sentiment Analysis* (Pang and Lee 2008) or *Semantic Textual Similarity* (Agirre *et al.* 2012). These word vectors (or **word embeddings**) are constructed following the distributional hypothesis, where the meaning of a word is learnt based on its context. They capture the distributional syntactic and semantic information based on the word co-occurrence statistics on large corpora (Mikolov *et al.* 2013b). Once we have an embedding for each word of our vocabulary we can estimate the similarity of any two words by computing the cosine between word vectors. For the first two layers we used the cosine similarity

between Collobert and Weston word vectors<sup>1</sup> (Collobert and Weston 2008) and cosine similarity between Mikolov word vectors<sup>2</sup> (Mikolov *et al.* 2013b. (Socher *et al.* 2011a) used Collobert and Weston word vectors for Paraphrase Detection, computing the similarity score as the Euclidean distance between the word vectors, with excellent results. Their system generates a probability for two sentences, and they assign the label Paraphrase if this probability is 0.5 or higher. Given that paraphrase and STS are closely related tasks and Collobert and Weston word vectors did well in Paraphrase Detection, using them for STS is a very good starting point. In addition, we used Mikolov word vectors to provide similar knowledge but from different corpora and a different method. Our hypothesis is that both word vectors are complementary.

Recent works also propose to encode word vectors using the structure stored in **knowledge bases** (e.g. WordNet) (Goikoetxea *et al.* 2015). In this model, the meaning of a word is encoded using random walks over the knowledge base, where each random walk generates an artificial context for a given word. Then, this artificial context is used to feed a *Neural Network Language Model* which is able to learn word vectors. These word vectors differ from the previous two because they encode the meaning based on a structured knowledge base, made by human experts, instead using unlabeled corpora. Wordnet based word vectors<sup>3</sup> provide more precise knowledge, but less coverage.

In the case of **Wikipedia**, we can use the Wikipedia dictionary (see UKB package<sup>3</sup>) to detect entities present in the sentences and measure their mutual similarity (Yeh *et al.* 2009). This dictionary is a list of strings with their corresponding entities and frequencies. We identify all strings appearing as an entry in our dictionary scanning the document tokens from left to right and consider the longest possible span which has a dictionary entry as a candidate mention. This method detects all strings appearing in the dictionary, such as *'titanium'* or *'house'*, not only entities. For example, in the example on Fig. 4.7 the system will select both *'Bill\_De\_Blasio'* and *'New\_York'* as candidate mentions. It would not consider *'De\_Blasio'* (there is already a longer match) nor *'as'* (it has no entry in the dictionary). For the similarity score between this entries we use the *Jaccard Similarity Coefficient* (see below) using the entities of the first and second strings as the sets A and B respectively:

---

<sup>1</sup><http://metaoptimize.com/projects/wordreprs/>

<sup>2</sup><code.google.com/p/word2vec>

<sup>3</sup><http://ixa2.si.ehu.es/ukb/>



$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (4.1)$$

- Bill De Blasio sworn in as New York mayor, succeeding Bloomberg
- Bill\_De\_Blasio sworn\_in New\_York mayor succeeding Bloomberg

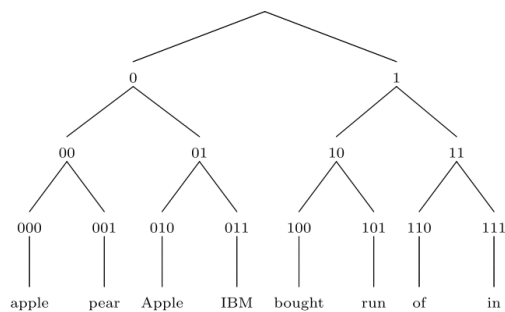
**Figure 4.7** – Example of matching Wikipedia entries. The above sentence is the original, and the second is the result after entities are detected. The words that form an entity are joined using underscores ('\_'), as in the case '*Bill\_De\_Blasio*' and '*New\_York*'.

One phenomena that happens when working with word vectors is that **numbers** are codified in very similar vectors, because they tend to appear in a identical context. If we try to measure the similarity between 1,000 and 4 using word vectors we will find that they almost mean the same. But '*4 dead in a car crash*' is not the same as '*1,000 dead in a car cash*'. To fix this issue we use another layer to deal with numbers. This layer will be a sparse matrix, where only the cells comparing numbers will have a value. We first need to identify the numbers contained in both sentences and, if necessary, transform them from text to normalized numbers. Then, the similarity between two numbers is defined following (Intxaurreondo *et al.* 2015) as:

$$Sim(num_1, num_2) = 1 - \frac{|num_1 - num_2|}{max(num_1, num_2)} \quad (4.2)$$

Words can also be represented by **clusters**. (Brown *et al.* 1992) used a hierarchical algorithm that maximize mutual information of bigrams to generate word clusters. (Clark 2003) made use of distributional and morphological information to gather morphologically similar words in the same cluster. (Agerri and Rigau 2016) induced clusters from these word vectors by applying *K-means clustering*. In their final data, similar words are clustered together. We can use clusters to fill more layers, assigning 1 if both words in each of the sentences are in the same cluster, and 0 otherwise. We use these three different clusters generated from different sources or corpora to add three more layers to the cube: *Word2vec clusters*, *Clark clusters* and *Brown clusters*. In the case of Brown clusters, the clustering algorithm is a hierarchical algorithm that produces a hierarchical clustering of the words, usually represented as a binary tree. In these trees, the path from the root

to a leaf (a word) is represented as a bit string, and choosing only a given number of bits on these path allows us to choose a different level of abstraction. In other words, if we choose the full path we are selected a more precise and smaller cluster, and if we choose a subset of bits we are selecting an intermediate node of the tree. For instance, in Fig. 4.8 we can choose '00' if we want to gather 'apple' and 'pear' in the same cluster. Based on training data, we decided to use paths of length 17.



**Figure 4.8** – A Brown clustering hierarchy.

In summary, the final cube is formed by the following layers:

1. Distributional: Cosine similarity between Collobert and Weston Word Vectors.
2. Distributional: Cosine similarity between Mikolov Word Vectors.
3. WordNet: Cosine similarity between word vectors derived from WordNet using random walks.
4. Wikipedia: Jaccard Similarity Coefficient of the entities present in Wikipedia.
5. Numbers: Heuristic similarity score for numbers.
6. Word2vec clusters: Similarity based on words sharing same cluster.
7. Clark clusters: Similarity based on words sharing same cluster.
8. Brown clusters: Similarity based on words sharing same cluster, using different abstraction levels.

The first three layers produce a dense layer, where most of the cells have a value. The other ones produce a sparse layer, where only the pairs occurring in the resource have a value, and the rest are unattested. In addition to their particular knowledge, all layers align tokens or lemmas that are exactly the same, with a score of 1.0. The exception is the Numbers layer, which only aligns numbers, and ignores the rest. The density of each of the layer in shown in Table 4.1.

Method	MSRpar12 Train	MSRvid12 Train	HDL13-14	Images14	Mean
Collobert	80.49	83.25	77.60	82.20	80.89
Mikolov	43.35	41.40	44.89	46.24	43.97
WordNet	26.82	36.73	45.99	30.89	35.10
Wikipedia	4.45	10.26	9.89	8.12	8.18
Numbers	0.29	0.03	0.40	0.12	0.21
Clusters (Word2vec)	7.01	14.29	10.32	9.93	10.39
Clusters (Clark)	4.32	10.59	7.24	8.36	7.63
Clusters (Brown)	4.09	10.50	6.80	8.38	7.44

**Table 4.1** – Layer density by dataset of train and development data (percentage of cells with a non-zero value).

### Discarded Layers

The final layers of the cube are the result of a development process. In this development, other layers were tested on train and development data and discarded. For completeness we also report them here.

The **Paraphrase Database** (PPDB) (Ganitkevitch *et al.* 2013) contains over 220 millions of paraphrase pairs (both words and phrases). The paraphrases are extracted from bilingual parallel corpora (more than 100 million sentence pairs and more than 2 billion words). Each pair includes a probability for one word/phrase to be a paraphrase of the second word/phrase. These probabilities can be easily used as similarity scores. PPDB yields conditional probabilities. As our scores are undirected, in case the database contains values for both directions, we averaged both numbers. We tried the XXXL version using the *lexical paraphrases*, *One-To-Many*, *Many-To-One* and the *Phrasal Phrases*. Few of the paraphrases were found in our data, and therefore the layer hardly affected the final results. This fact, together with the high computational and memory cost, led to discard this layer.

(Resnik 1995) presented a measure of semantic similarity metric based on the notion of **information metric**. We added this metric to the cube, but when executing the system we did not get any improvement. This may be caused because the provided information is already implicit within the cube through WordNet word vectors.

We think that the final similarity score between two sentences is specially based in the similarities between words/phrases that humans discern in those sentences, and not on dissimilarities. This idea of giving more importance to similarity (to higher values) than to dissimilarity (lower values) is important for some decisions that are made in the next section. Nevertheless, we also considered the addition of a **penalty layer**. To do so we used the dataset and the neural network model presented in (Kruszewski and Baroni, 2015) to create word vectors for our vocabulary that perform compatibility detection. This layer was designed to penalize cells where the words were detected as incompatible. The results were not good enough, and the layer was discarded, although it is expected to incorporate it in the future. As the layer was discarded, no analysis was carried out to support this idea.

## 4.3 Producing the STS Score

At this point we have constructed a cube using information from several sources. The next step is to use the information stored in the cube to produce a single number for each of the pairs. In the next section we explain the formula used to extract a similarity score from our cube. Right after we explain the technique of the threshold (Section 4.3.2) and a variation of the cube, the hierarchical cube (Section 4.3.3), which allowed to improve the performance during the designing step. Finally we describe the ML features in Section 4.3.4. The threshold, the hierarchical cube and ML features are optional (we can get a score without them), unlike the scoring step, which is mandatory.

### 4.3.1 Pairwise similarity score

(Mihalcea *et al.* 2006) presented a method for measuring the semantic similarity of texts as a function of the semantic similarity of the components words. They did this by combining metrics of word-to-word similarity (similarity scores for a given word pair) and word specificity (IDF scores for the words, see below) in a formula capable to assign a good semantic similarity score for to input sentences  $S_1$  and  $S_2$  :

---

### 4.3. PRODUCING THE STS SCORE

$$sim(S_1, S_2) = \frac{1}{2} \left( \frac{\sum_{w_i \in S_1} (idf(w_i) * \max_{w_j \in S_2} Sim(w_i, w_j))}{\sum_{w_i \in S_1} idf(w_i)} + \frac{\sum_{w_j \in S_2} (idf(w_j) * \max_{w_i \in S_1} Sim(w_j, w_i))}{\sum_{w_j \in S_2} idf(w_j)} \right) \quad (4.3)$$

where *Inverse Document Frequency* (IDF) is an inverse function of the number of documents in which that term occurs that can be used to quantify the specificity of a term/word.

To produce a similarity score using our cube we extend the pairwise similarity scoring function presented in (Mihalcea *et al.* 2006) to work with several dimensions, where each dimension reflects a source of information. Following our hypothesis (see Section 4.1), for each word in a sentence we search for the maximum similarity value with a word in the other sentence across all the layers. (Kusner *et al.* 2015) used a similar approach by matching directly to the nearest neighbour (the most similar), ignoring the other neighbours. To compute IDF values we used the frequency lists from *Brown and LOB corpora* of written English<sup>4</sup>. If a word is not in the IDF list we assign to it the highest IDF (it is considered that provides the highest information). Once we have constructed the cube and the IDF scores, we compute the final STS score using the new scoring function:

$$sim(S_1, S_2) = \frac{1}{2} \left( \frac{\sum_{w_i \in S_1} (idf(w_i) * \max_{d \in D, w_j \in S_2} (\alpha_d * Sim(d, w_i, w_j)))}{\sum_{w_i \in S_1} idf(w_i)} + \frac{\sum_{w_j \in S_2} (idf(w_j) * \max_{d \in D, w_i \in S_1} (\alpha_d * Sim(d, w_j, w_i)))}{\sum_{w_j \in S_2} idf(w_j)} \right) \quad (4.4)$$

where  $\alpha_d$  represents a **weighting value** for each of the layers ( $d$  is a layer and  $D$  is the total number of layers). This weighting is necessary because the confidence of the different resources may not be the same (and it's not). These weights represent the confidence in each of the layers/resources. For instance, if a layer with a weighting score of 0.3 assigns a similarity score of 0.8 ( $0.8 * 0.3 = 0.24$ ) between two words and other layer with a weighting score of 0.9 assigns a 0.6 ( $0.93 * 0.6 = 0.54$ ) the final similarity for those words will be 0.54. We set the values of those free parameters (one per layer) in train and development data.

---

<sup>4</sup>[http://sslmit.unibo.it/~baroni/brown\\_lob\\_fq\\_lists.html](http://sslmit.unibo.it/~baroni/brown_lob_fq_lists.html)

### 4.3.2 Threshold

As we have seen earlier, some resources generate a completely filled matrix, while others create sparser matrices, where not all cells have a value. But the fact of having a value does not guarantee that this value is appropriate or correct. Some resources always yield a value however how different words may be between them. In some cases it does not even make sense to relate these concepts. This effect is most noticeable with word vectors, which can give values above zero even for concepts that any human would find completely unrelated. Although this is not always the case, most of the values below a threshold could be considered as noise. Since it is not desirable to increase the value of similarity due to this noise, we apply a threshold below which all values become zero.

Preliminary tests applying this threshold substantially improved the final correlation obtained with the cube, so we decided to incorporate it into the similarity formula. However, to go one step further we decided to use this threshold as a limit on which a '*bad alignment*' has been chosen. To do this, we look at whether the maximum similarity value for a word on the first sentence with respect to the words on the second sentence is less than the chosen threshold. This procedure is intended to penalize bad alignments (something similar was tested in (Han *et al.* 2013)). This way, we now have two summations, one that we will call '*Positive*' and another '*Negative*', and also the two summations of the IDF values. Therefore, the scoring function is now defined as follows.

Given a similarity between two tokens  $w_i \in S_1$  and  $w_j \in S_2$  we define:

$$sim(w_i, w_j) = idf(w_i) * \max_{d \in D, w_i \in S_1, w_j \in S_2} (\alpha_d * sim(d, w_i, w_j)) \quad (4.5)$$

where  $\alpha_d$  represents a **weighting value** for the layer  $d$  among our collection of layers  $D$ , as seen in formula 4.4, we define two similarity formulas for sentences:

$$positive(S_1, S_2) = \left( \sum_{w_i \in S_1} sim(w_i, w_j) \text{ if } \max_{w_j \in S_2} (sim(w_i, w_j)) \geq threshold \right) + \left( \sum_{w_j \in S_2} sim(w_j, w_i) \text{ if } \max_{w_i \in S_1} (sim(w_j, w_i)) \geq threshold \right) \quad (4.6)$$

### 4.3. PRODUCING THE STS SCORE

---

$$\begin{aligned}
 \text{negative}(S_1, S_2) = & \left( \sum_{w_i \in S_1} \text{sim}(w_i, w_j) \text{ if } \max_{w_j \in S_2} (\text{sim}(w_i, w_j)) < \text{threshold} \right) + \\
 & \left( \sum_{w_j \in S_2} \text{sim}(w_j, w_i) \text{ if } \max_{w_i \in S_1} (\text{sim}(w_j, w_i)) < \text{threshold} \right) \quad (4.7)
 \end{aligned}$$

Formula 4.6 is a summation of the alignments from the first sentence ( $S_1$ ) to the second sentence ( $S_2$ ) and of the alignments from  $S_2$  to  $S_1$  where the similarity between these word pairs is above the threshold. Formula 4.7 is the summation of the alignments where the similarity is below the threshold. For example, suppose that the third word of  $S_1$  ( $S_{1_3}$ ) has its maximum similarity with the fourth word of  $S_2$  ( $S_{2_4}$ ), and that this similarity is 0.6 ( $\text{sim}(S_{1_3}, S_{2_4}) = 0.6$ ). If the threshold is 0.5, this similarity will be added to  $\text{positive}(S_1, S_2)$  (Formula 4.6), and otherwise to  $\text{negative}(S_1, S_2)$  (Formula 4.7). This is done for every word of  $S_1$  with its maximum similarity to words of  $S_2$  (and vice versa).

We also define two formulas to measure the IDF of the sentences:

$$\begin{aligned}
 \text{positive\_idf}(S_1, S_2) = & \left( \sum_{w_i \in S_1} \text{idf}(w_i) \text{ if } \max_{w_j \in S_2} (\text{sim}(w_i, w_j)) \geq \text{threshold} \right) + \\
 & \left( \sum_{w_j \in S_2} \text{idf}(w_j) \text{ if } \max_{w_i \in S_1} (\text{sim}(w_j, w_i)) \geq \text{threshold} \right) \quad (4.8)
 \end{aligned}$$

$$\begin{aligned}
 \text{negative\_idf}(S_1, S_2) = & \left( \sum_{w_i \in S_1} \text{idf}(w_i) \text{ if } \max_{w_j \in S_2} (\text{sim}(w_i, w_j)) < \text{threshold} \right) + \\
 & \left( \sum_{w_j \in S_2} \text{idf}(w_j) \text{ if } \max_{w_i \in S_1} (\text{sim}(w_j, w_i)) < \text{threshold} \right) \quad (4.9)
 \end{aligned}$$

Formulas 4.8 and 4.9 are formulas to compute the IDF of the sentences. The conditions are the same as for  $\text{positive}(S_1, S_2)$  (Formula 4.6) and  $\text{negative}(S_1, S_2)$  (Formula 4.7), but instead of adding the values of  $\text{sim}(w_i, w_j)$  (Formula 4.5) for each alignment above or below the threshold we sum the IDF score ( $\text{idf}(w_i)$ ) of the words in  $S_1$  and  $S_2$ .

Using these partial scores we tested the performance of the threshold using the 5 alternative formulas shown in Table 4.2. **Convert to 0** is a formula that discards

## CHAPTER 4. CUBES FOR SEMANTIC TEXTUAL SIMILARITY

Method	Description	Formula
Convert to 0	The values below the threshold are converted to zero.	$\frac{positive(S_1, S_2)}{positive\_idf(S_1, S_2) + negative\_idf(S_1, S_2)}$
Add and subtract	The values above the threshold are added, and the values below are subtracted.	$\frac{positive(S_1, S_2) - negative(S_1, S_2)}{positive\_idf(S_1, S_2) + negative\_idf(S_1, S_2)}$
Only positives	The values above the threshold are added, and the values below are ignored. The IDF value of values below the threshold is not computed (it's not added in the denominator).	$\frac{positive(S_1, S_2)}{positive\_idf(S_1, S_2)}$
Only negatives	The values below the threshold are added, and the values above are ignored. The IDF value of values above the threshold is not computed (it's not added in the denominator).	$\frac{negative(S_1, S_2)}{negative\_idf(S_1, S_2)}$
Subtraction	We subtract the result of 'Only negatives' to the results of 'Only positives'.	$\frac{positive(S_1, S_2)}{positive\_idf(S_1, S_2)} - \frac{negative(S_1, S_2)}{negative\_idf(S_1, S_2)}$
Proportional subtraction	As 'Subtraction', but we subtract more in terms of how far away is the value from the threshold. Thus, if the value is very close to the threshold, we subtract a very small value.	$\frac{positive(S_1, S_2)}{positive\_idf(S_1, S_2)} - \left( \frac{prop\_negative(S_1, S_2)}{negative\_idf(S_1, S_2)} \right)$

**Table 4.2** – Different methods for applying a threshold in the pairwise similarity scoring.

everything that is below the threshold (does not sum up this scores). This method can be used to filter the noise that may exist in some resources, removing residual similarity scores. **Add and subtract** adds to the summation the similarities that are above the threshold, and subtracts the ones that are below. **Only positives** takes into account those alignments whose value is above the threshold and ignores the rest, including the IDF value of those words. **Only negatives** does the same as *Only positives*, but with those that are below the threshold. **Subtraction** is the result of subtracting *Only negatives* to the value of *Only positives*. Finally, **Proportional subtraction** is a method to subtract less value if the score is near the threshold, and more if this score is far below the threshold (similarities above the threshold are summed up). After testing these methods on the training data, we decided to use the *Proportional subtraction* method. Thus,  $negative(S_1, S_2)$  (Formula 4.7) is substituted by  $prop\_negative(S_1, S_2)$  (Formula 4.10), which is defined as follows:



### 4.3. PRODUCING THE STS SCORE

---

$$\begin{aligned} \text{prop\_negative}(S_1, S_2) = & \left( \sum_{w_i \in S_1} \text{prop\_sim}(w_i, w_j) \text{ if } \max_{w_j \in S_2} (\text{sim}(w_i, w_j)) < \text{threshold} \right) + \\ & \left( \sum_{w_j \in S_2} \text{prop\_sim}(w_j, w_i) \text{ if } \max_{w_i \in S_1} (\text{sim}(w_j, w_i)) < \text{threshold} \right) \end{aligned} \quad (4.10)$$

where  $\text{prop\_sim}(w_i, w_j)$ , the similarity between two words  $w_i \in S_1$  and  $w_j \in S_2$ , is now defined as follows:

$$\text{prop\_sim}(w_i, w_j) = \text{idf}(w_i) * |\text{threshold} - \max_{d \in D} (\alpha_d * \text{Sim}(d, w_i, w_j))| \quad (4.11)$$

where  $\alpha_d$  represents a **weighting value** for the layer  $d$  among our collection of layers  $D$  as seen in Formula 4.5. Using this formula the values that are below the threshold subtract more the more lower they are. In other words, if they are below the threshold but only by a close margin they are not considered as completely bad alignments, and they almost penalize.

#### 4.3.3 Hierarchical cube

A side effect of using weighting values for each of the layers is that we may need to choose a very low score for some layers just to be able to choose the most accurate score among the layers. If the 'correct' similarity between two words is given by the layer based on WordNet but this score is low (for instance, 0.4) and another layer has a score of 0.9 we need the weighting value of the second layer to be 0.4 or lower. Layers based on knowledge bases (such as WordNet and Wikipedia) assign more accurate values, despite the fact that they produce more sparse layers. In order to obtain the best results we are constraining the possible values for the weighting scores of the layers.

A Hierarchical cube solves this issue dividing the cube in two different levels. The upper levels includes the most accurate and reliable layers: all except the distributional layers, based on Collobert and Weston and Mikolov word vectors. This layers are usually more sparse but their scores are trustworthy, specially if they are specialized (such as numbers). We use the same approach used in the previous section, selecting the highest score (after weighting) for each pair of words among the layers on the upper level. If we don't have a score for a given pair of words, we choose the highest score (after weighting) among the layers of the lower levels. This approach allows us to choose high weighting values for

the layers on the lower level while still selecting the more accurate values of the layers on the upper level.

#### 4.3.4 Machine Learning

Apart from all the knowledge stored in the cube, there are other features or characteristics of the sentences, which is important to model. Therefore, we generate ML features that we will use to feed a Linear Regressor.

Once we obtain the similarity score from the cube using the pairwise similarity formula we feed a Linear Regressor with this number and other features from other knowledge sources not present in the cube. These features include 8 features extracted using the *ROUGE package* (Lin 2004), 6 features for numbers, 3 features for *Named Entities*, 3 features for the length of the sentences, and 10 features from *Interpretable STS* (Lopez-Gazpio et al. 2017).

**ROUGE** is a package for automatic evaluation of summaries. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans (Lin 2004). Instead of using an automatically generated summary and comparing it to another one created by humans we used the ROUGE package to measure the similarity between two sentences. We use one sentence as an automatically generated sentence and the second one as the human created sentence and run the system. The following features were selected and generated employing the ROUGE package (Lin 2004):

- Rouge-1.
- Rouge-2.
- Rouge-3.
- Rouge-4.
- Rouge-L.
- Rouge-W1.2.
- Rouge-S\*.
- Rouge-SU\*.

### 4.3. PRODUCING THE STS SCORE

---

Even if we have a layer for **numbers** we added five additional number features because numbers are very relevant for STS. A difference between numbers in two sentences can drastically modify the similarity, changing them from being equivalent to something that can be a contradiction, leading the annotators to score them with a very low value. We fed the Linear Regressor with the following features:

- Number of numbers in the first sentence.
- Number of numbers in the second sentence.
- Absolute difference.
- (Boolean) 1 if both sentences contain exactly the same numbers or they don't contain any number, and 0 otherwise.
- (Boolean) 1 if both sentences contain exactly the same numbers, and 0 otherwise.
- (Boolean) 1 if the numbers in one sentence are a subset of the numbers in the other sentence, and 0 otherwise.

In the same way than numbers can change the similarity between two sentences, this can also happen with **named entities**. Even if we handle this with the Wikipedia layer we add features to control if there is an excessive difference in the number of entities in the two sentences. We do the same with the **length** of the sentences, because a big difference may suggest that there is missing information. The last six features are the following:

- Number of entities in the first sentence.
- Number of entities in the second sentence.
- Absolute difference in the number of entities.
- Number of tokens in the first sentence.
- Number of tokens in the second sentence.
- Absolute difference in the number of tokens.

**Interpretable STS** (iSTS) is a task which goal is to provide an explanatory layer to regular STS. We use featured from iSTS to feed the ML system:

- Number of aligned segments.
- Number of segments labeled as **Equivalent**.
- Number of segments labeled as **Specific**.
- Number of segments labeled as **Similar**.
- Number of segments labeled as **Related**.
- Number of segments labeled as **Opposite**.
- Number of segments labeled as **Equivalent**, **Specific** or **Similar** with an score of **4 or higher**.
- Number of segments labeled as **Equivalent**, **Specific** or **Similar** with an score of **3 or higher**.
- Number of segments labeled as **Equivalent**, **Specific** or **Similar** with an score of **2 or higher**.
- Average score among aligned segments.

## 4.4 Evaluation

In this section we will evaluate the cube. Its performance will be compared with another method in which the cube is not used globally, but using each layer individually. The cube will also be evaluated against the best STS systems.

Among all the datasets created for STS, and introduced in Section 3.3, we have selected only those that present a more natural language. The sentences in this datasets are more natural than in other datasets such as OnWN, FnWN, or SMT (glosses from OntoNotes-WordNet and FrameNet-Wordnet respectively, and sentences from Machine Translation evaluation), which include automatically translated sentences. We evaluated our system on the MSR Paraphrase (paraphrases from news sources), MSR Video (descriptions of short videos provided by annotators), Headlines (headlines from real news, present in the 2013 and 2014 task) and Images (captions of images, present in the 2014 task) datasets. We divided the datasets into train, development and test as shown in Table 4.3.

When splitting a dataset into two parts, it has been taken into account to maintain the distribution of the values. Thus, in each half there is a similar quantity of similarities for each range ([0-1], ..., (4-5]).

	Name	Description	Pairs
Train	MSRpar12 Train	MSR Paraphrase train set (STS 2012).	750
	MSRvid12 Train (50%)	50% of MSR Video train set (STS 2012).	375
	HDL13	Headlines dataset (STS 2013).	750
	Images2014 (50%)	50% of Images dataset (STS 2014).	375
Dev.	MSRvid12 (50%)	50% of MSR Video train set (STS 2012).	375
	HDL14	Headlines dataset (STS 2014).	750
	Images2014 (50%)	50% of Images dataset (STS 2014).	375
Test	MSRpar12 Test	MSR Paraphrase test set (STS 2012).	750
	MSRvid12 Test	MSR Video test set (STS 2012).	750
	HDL15	Headlines dataset (STS 2015).	750
	HDL16	Headlines dataset (STS 2016).	249
	Images2015	Images dataset (STS 2015).	750

**Table 4.3** – Data split into Train, Development and Test sets.

To evaluate the performance of the system we use the official scorer provided by the STS organizers, which computes the *Pearson Correlation* score between the system scores and the *Gold Standard* scores (see Section 3.5.1).

#### 4.4.1 Train

Once we have built the cube and designed the scoring function we used the training set to measure its performance and select the parameters. As a first step, we evaluate each resource individually on Training data (as if the cube only had one layer) to estimate the minimum performance for the cube, and to see the reliability of each resource. In this phase it is perceived that the distribution of similarity values from different resources is not the same. For example, according to *Jaccard* over Wikipedia a similarity value of 0.2 is a very high value, since this value is within the 5% of highest values. To solve this, we adjust some curves, changing the values of the resources, so that they individually obtain the highest possible correlation (see table 4.4 for adjustments). Once each resource has been adjusted, the optimal parameters are searched using Grid-search to assign the weights for each layer, as well as the threshold that produce the best results (the final parameters are in the table 4.7). In the case of individual layers it does not make sense to weight the layer, but we make use of the threshold.

The next part of the development consisted on testing on the training set all the possible configurations presented in the previous sections. These configurations

Layer		
Collobert	Mikolov	Wikipedia
$sim = sim^3$	$sim = sim^3$	$sim = \sqrt[5]{sim}$

**Table 4.4** – Curve adjustment for layers.

Method	MSRpar12 Train	MSRvid12 Train (50%)	HDL13	Images14 (50%)	Mean
Collobert	0.436	0.643	0.561	0.656	0.574
Mikolov	0.341	0.615	0.325	0.656	0.484
WordNet	0.682	0.805	0.755	0.800	0.760
Wikipedia	0.663	0.774	0.772	0.762	0.742
Numbers	-0.120	0.066	0.089	0.063	0.024
Clusters (W2v)	0.634	0.581	0.666	0.706	0.647
Clusters (Clark)	0.656	0.769	0.747	0.762	0.734
Clusters (Brown)	0.659	0.637	0.728	0.704	0.682
LR (I)	0.684	0.803	0.767	0.803	0.764
LR (I+F)	0.719	0.814	0.790	0.816	0.785
Hierarchical Cube	0.691	0.826	0.784	0.816	0.779
LR (I+C)	0.694	0.826	0.783	0.822	0.781
LR (C+F)	<b>0.725</b>	0.828	0.799	0.811	0.791
LR (I+C+F)	<b>0.725</b>	<b>0.830</b>	<b>0.804</b>	<b>0.829</b>	<b>0.797</b>

**Table 4.5** – Results of our system on Train data using 10-fold cross validation. LR is a *Linear Regressor* fed with different scores and features: scores from Individual layers (I), the cube score (C), and ML features (F).

include testing and refining the similarity function (Section 4.3.1), tests to choose the best method to use the threshold (Section 4.3.2), compare the initial cube with the hierarchical cube (Section 4.3.3), and design and discard the features for ML (Section 4.3.4). These tests on the training set have led to the decisions and the final design presented in Sections 4.2.1 and 4.3. All these partial results are computed using *10-fold cross validation*.

Once these decisions were made and the final design of the system was defined, for each individual layer we obtain the correlations reflected in the table 4.5, being the best resources WordNet and Wikipedia. The result of the cube is reflected in the line '*Hierarchical Cube*', improving the best individual result by almost 2 points. But in order to compare this value we need to compare it with a

system that has access to the same knowledge and the same information. Using a linear regressor<sup>5</sup> we combine the outputs of all the individual layers obtaining the result of the line ' $LR(I)$ '. The cube gets a higher score, almost 2 points better, without using ML. In the table 4.5 there are additional results using the features for Machine Learning presented in Section 4.3.4. The legend used on table 4.5 for the linear regressor is as follows:

- (I): Fed with the scores of all the individual layers.
- (I+F): Fed with the scores of all the individual layers plus ML features.
- (I+C): Fed with the scores of all the individual layers and the scores from the cube.
- (C+F): Fed with the scores of the cube plus ML features.
- (I+C+F): Fed with the scores of all the individual layers, the scores from the cube, and ML features.

All the features we incorporate to the linear regressor improve the correlation, demonstrating that they are useful and complementary to the previous ones. The best result using the cube is the one obtained by the ' $LR(I+C+F)$ ', which on average obtains a result 1.2 higher than the best result that does not use the cube, ' $LR(I+F)$ '.

## 4.4.2 Development

In the previous section we have created the preliminary system, whose evaluation has been done using cross-validation. To validate those results, the next step is to use the Development set as Test set. Using the values obtained training on the Train set for the different parameters of the cube, we evaluate the system on Development set. We used the values chosen for the first half of MSRvid and Images on the second half, and the values obtained for HDL13 on HDL14.

Table 4.6 shows the development results. The values of the parameters have not been adjusted on the datasets itself, and in spite of this the results are maintained. The best result with the cube (' $I+C+F$ ') gets 1.9 points of improvement in respect to the best result without the cube (' $I+F$ '). The results in general are a reflection of those obtained on the train using cross-validation. This proves that this system is viable and has a good performance, where each of the features improves the previous result. The next step is to merge the training and development sets. With this larger dataset we find again the optimal values of the parameters. Being the training set larger, the values obtained will be more stable to use them on the Test set.

<sup>5</sup>Weka implementation for Linear Regressor (LR), without attribute selection.

Method	MSRvid12 Train (50%)	HDL14	Images14 (50%)	Mean
Collobert	0.726	0.552	0.636	0.638
Mikolov	0.638	0.321	0.590	0.516
WordNet	0.841	0.718	0.755	0.771
Wikipedia	0.821	0.762	0.713	0.765
Numbers	0.079	0.052	0.041	0.057
Clusters (W2v)	0.653	0.646	0.649	0.649
Clusters (Clark)	0.826	0.707	0.720	0.751
Clusters (Brown)	0.683	0.696	0.679	0.686
LR (I)	0.839	0.759	0.766	0.788
LR (I+F)	0.838	0.783	0.793	0.804
Hierarchical Cube	0.853	0.758	0.790	0.800
LR (I+C)	0.854	0.763	0.800	0.805
LR (C+F)	0.856	0.777	0.822	0.818
LR (I+C+F)	<b>0.858</b>	<b>0.785</b>	<b>0.827</b>	<b>0.823</b>

**Table 4.6** – Results of our system trained on Train data and tested on Development data. LR is a Linear Regressor fed with different scores and features: scores from Individual layers (I), the cube score (C), and ML features (F).

The training data for MSRpar and MSRvid is going to be the Train datasets (full datasets) from STS2012 and will be evaluated on the Test dataset for MSRpar and MSRvid of the same year. HDL13 and HDL14 are put together and the optimum values obtained for them will be used to evaluate on HDL15 and HDL16. Finally, the values trained on Images14 (full dataset) will be used to evaluate on Images15. The results (omitted for brevity) are similar to the previous ones, obtaining an improvement of 1.7 points thanks to the cube (*'Lin. Regressor (I+C+F)'*) with respect to the best result without using it (*'Lin. Regressor (I+F)'*).

The optimal parameters found in this step are shown in Table 4.7, which turned out to be quite similar to the ones obtained with the train data, with only small variations. The weights for WordNet and Wikipedia are always high, indicating that their knowledge is always useful and accurate. Other resources obtain very different weights for the different datasets. For instance, Clark clusters' weight is high for MSRvid and HDL datasets, but low for MSRpar and Images. Collobert word vectors are quite useful for HDL, but they are not selected for MSRvid and Images datasets. We are going to use these optimal parameters to run the system on Test data in the next section.



	MSRpar	MSRvid (50%)		HDL13	HDL13-14	Images14 (50%)	
		MSRvid				Images14	Images14
Collobert	0.2	0.0	0.0	0.4	0.3	0.0	0.0
Mikolov	0.6	0.5	0.6	0.5	0.3	0.5	0.6
WordNet	1.0	0.9	0.8	0.9	0.7	0.8	0.8
Wikipedia	1.0	0.9	0.8	1.0	1.0	0.8	0.7
Numbers	0.0	0.9	0.8	0.3	0.3	0.5	0.4
Clusters (W2v)	0.4	0.0	0.0	0.1	0.1	0.0	0.0
Clusters (Clark)	0.4	0.8	0.8	1.0	1.0	0.3	0.4
Clusters (Brown)	0.7	0.0	0.0	0.1	0.4	0.0	0.0
Threshold	0.0	0.1	0.1	0.1	0.0	0.1	0.1

**Table 4.7** – Final selection of parameters based on Train+Development data.

### 4.4.3 Test

Once the system is optimized with the values of the table 4.7, we execute it on the Test set. For MSRpar and MSRvid test sets we train on MSRpar and MSRvid train data, for HDL15 and HDL16 we train on HDL13 and HDL14 (merged), and for Images15 we train on Images14. The results are shown in the table 4.8. The differences in this table (those lines in italic and starting with a  $\Delta$ ) are multiplied by 100, to appreciate them better (we do this with any other table as well). The results that were observed in table 4.6 are reflected in the results obtained for the test set, and each of the features and characteristics employed improves the correlation.

Our Hierarchical Cube gets a mean correlation of 0.789, which is comparable to the result obtained by the linear regressor fed with the scores from the individual layers. It is important to emphasize that we are comparing a system that uses ML against another one that does not use ML, and still the cube is comparable to the linear regressor model (line ' $\Delta LR (I)$ ' on table 4.8). When we compare the best linear regressor model fed with the cube against the best linear regressor model trained without the cube we see that the cube overperforms the last one by 1.48 points (line ' $\Delta LR (I+F)$ ' on table 4.8). This demonstrates that we could extract more knowledge of our resources thanks to the cube.

Method	MSRpar12 Test	MSRvid12 Test	HDL15	HDL16	Images15	Mean
Collobert	0.392	0.705	0.603	0.523	0.691	0.583
Mikolov	0.308	0.621	0.441	0.362	0.695	0.485
WordNet	0.608	0.849	0.796	0.790	0.839	0.776
Wikipedia	0.591	0.818	0.812	0.800	0.811	0.766
Numbers	-0.092	0.043	0.092	0.011	0.193	0.049
Clusters (W2v)	0.568	0.697	0.741	0.722	0.726	0.691
Clusters (Clark)	0.580	0.821	0.783	0.771	0.811	0.753
Clusters (Brown)	0.581	0.682	0.765	0.752	0.753	0.706
LR (I)	0.626	0.857	0.814	0.804	0.847	0.790
LR (I+F)	0.694	0.851	0.814	0.807	0.833	0.800
Hierarchical Cube	0.615	0.863	0.816	0.805	0.849	0.789
$\Delta LR (I)$	-1.13	+0.60	+0.16	+0.07	+0.20	-0.02
LR (I+C)	0.630	0.873	0.822	0.812	<b>0.859</b>	0.799
LR (C+F)	0.685	0.846	0.816	0.807	0.838	0.798
LR (I+C+F)	<b>0.697</b>	<b>0.889</b>	<b>0.823</b>	<b>0.816</b>	0.847	<b>0.814</b>
$\Delta LR (I+F)$	+0.34	+3.78	+0.95	+0.84	+1.47	+1.48
Chimera-1	UKP-run2	0.683	0.874			
	DLS@CU-S1			0.825	0.864	0.815
	Samsung				0.828	
	$\Delta LR (I+C+F)$	+1.40	+1.46	-0.16	-1.20	-1.70
Chimera-2	Takelab-simple	0.734	0.880			
	Samsung-delta			0.842		
	Samsung-beta				0.871	0.831
	Samsung				0.828	
	$\Delta LR (I+C+F)$	-3.73	+0.82	-1.83	-1.20	-2.39

**Table 4.8** – Results of our system on Test data. LR is a Linear Regressor fed with different scores and features: scores from Individual layers (I), the cube score (C), and ML features (F).  $\Delta LR (I)$  is the difference between *Hierarchical Cube* and *LR (I)*.  $\Delta LR (I+F)$  is the difference between *LR (I+C+F)* and *LR (I+F)*.  $\Delta LR (I+C+F)$  is the difference between *LR (I+C+F)* in respect to the virtual *Chimera-1* and *Chimera-2* systems.

Regarding the state-of-the-art, we can't compare the cube with any system that competed all the years, and therefore we have created two chimera systems. *Chimera-1* is a mixture of the systems that won the competition in the year of release of the datasets (the system that obtained the highest mean correlation), and gathers the results of UKP-run2 (2012) (Bar *et al.* 2012), DLS@CU-S1 (2015) (Sultan *et al.* 2015) and Samsung (2016) (Rychalska *et al.* 2016). *Chimera-2* is a mixture of the systems that achieved the highest correlation in each of the datasets, and gathers Takelab-simple (2012) (Šarić *et al.* 2012), Samsung-beta (2015), Samsung-delta (2015) (Han *et al.* 2015) and Samsung (2016). These systems may be specialized in that particular dataset, and perform poorly on other datasets. Our system matches the performance of the *Chimera-1* system, and is only 1.67 point below the *Chimera-2* system. It is necessary to take into account that *Chimera-2* obtains the highest possible result derived from choosing the best system in each dataset. Still, our system is close to *Chimera-2*, demonstrating that the cube contributes enough to make our system a system state-of-the-art system.

#### 4.4.4 Ablation test

Finally, in Table 4.9 we present an ablation test. The differences listed in the 'Loss' column are shown multiplied by 100, to appreciate them better. In this table we can see that the losses derived from removing layers are generally small, with the exception of the WordNet layer, without which the systems result drops 2.5 points. This happens because the resources are complementary to each other, but there is also some overlap between them. This makes the cube very stable, since if one layer fails or is removed from the cube another is able to assume this loss. To check this fact we add three more lines:

- **Distributional:** We removed both Collobert and Mikolov layers, to see the effect of removing the lower level of the cube (and the two denser layers). When we remove this two layers at the same time, the system practically losses the sum of the losses of these layers separately.
- **Wn, Wiki:** We also remove WordNet and Wikipedia at the same time, because in the absence of one the other replaces it. As it is seen, removing Wordnet causes a loss of 2.5 points and removing Wikipedia a loss of 0.7 points, but removing both layers causes a loss of 3.9 points, which is more than the sum of both.
- **Cluster:** We also remove the three layers of clusters to see their contribution. In this case the loss is not as pronounced as in the two previous cases.

## 4.5 Conclusion

In this section we present a system for *Semantic Textual Similarity* based on the idea of a better combination of resources. We have constructed a cube of pairwise token similarities where each resource is added as a layer of this cube. Our hypothesis was that we can obtain better results **combining** word-to-word similarity from different sources at the word level, in contrast to other works where each resource is used independently. We investigated with several resources and after a selection process on train and development data we selected eight of them.

We have experimented with different metrics to extract a STS score from the cube. We used a well-known pairwise similarity scoring function and extended it to work with more than one dimension. We explored other ways to improve the system, using a thresholding technique to remove noise from the cube, and to detect bad alignments between candidate words and penalize them. We also studied the behaviour of the cube on the train data, and after we carried out this analysis we transformed the cube into a two-level hierarchical cube that improved the results. In addition to the cube we generated several features and trained a ML model using these features and the knowledge stored in the cube.

This system has obtained state-of-the-art results. The better results when using the cube show that our hypothesis was true, both with ML and without ML. The ablation test performed on the system demonstrated that the system is also very

Method	MSRpar12 Train	MSRvid12 Train	HDL13-14	Images14	Mean	Loss
Hierarchical Cube	0.691	0.841	0.775	0.804	0.778	-
- Collobert	0.687	0.840	0.755	0.802	0.771	-0.7
- Mikolov	0.688	0.837	0.770	0.795	0.773	-0.5
- WordNet	0.679	0.809	0.730	0.793	0.753	-2.5
- Wikipedia	0.685	0.840	0.756	0.802	0.771	-0.7
- Numbers	0.687	0.840	0.770	0.802	0.775	-0.3
- Cl. (Word2vec)	0.687	0.840	0.774	0.802	0.776	-0.2
- Cl.(Clark)	0.687	0.840	0.773	0.802	0.776	-0.2
- Cl. (Brown)	0.682	0.840	0.772	0.802	0.774	-0.4
- Distributional	0.689	0.837	0.745	0.795	0.767	-1.1
- WN, Wiki	0.670	0.806	0.700	0.777	0.738	-3.9
- Clusters	0.681	0.840	0.770	0.802	0.773	-0.5

**Table 4.9** – Ablation Test on Train+Development data.

strong and stable. Despite these good results, we believe that there are still more efficient methods to take better advantage of the knowledge stored in the cube. One of these methods may be to achieve the optimal alignments inside the cube by training an alignment selection algorithm.

A preliminary version of the cube was used in a system for *Interpretable STS* (Agirre *et al.* 2015b). Evaluation on iSTS is based on four criteria: alignments between chunks, type of alignment between chunks, similarity score between aligned chunks, and a combination of type and score. Similarity scores provided by the preliminary cube obtained the best performance among all submitted systems.



## Typed Similarity

In this chapter we introduce a new task related to *Semantic Textual Similarity* (STS). *Typed similarity* aims to identify the type of relation that holds between a pair of similar items in a digital library, and allows to provide an explanation of why items are similar, with applications to recommendation, personalisation and search.

We investigate the problem within the context of Europeana, a large digital library containing items related to cultural heritage. A set of similarity types was identified, and a set of 1500 pairs of items from the collection were annotated using crowdsourcing.

In the next chapter we present several approaches to automatically identifying the type of similarity that has been used in a real-world application.

### 5.1 Introduction

Nowadays there is a lot of cultural heritage material available through online portals. The immense amount of material can be overwhelming for users, spoiling their experience while exploring all these items. This negative sensation of user is increased if they do not receive any help in this exploration. This contrasts with the real world, where museums are organized by theme, or where you can consult the museum staff.

Search engines and digital libraries often allow users to search for similar items, an important function which supports exploratory search ([Marchionini 2006](#)) and sense-making ([Hearst 2009](#)). Users are often provided with similar items in

the form of a link from an individual item to a set of others in the collection. For example, Google Scholar<sup>1</sup> and PubMed<sup>2</sup>, both digital libraries containing academic publications, provide users with such links. Google Scholar has a link to 'Related Articles' and PubMed to 'Related Citations'. This feature is so important that it is implemented in many open-source search engines, e.g. Lucene and Terrier (Ounis *et al.* 2006; McCandless *et al.* 2010).

Similar items are normally identified using word-overlap measures. Following this approach, the similarity of a pair of documents is determined by counting the number of words they have in common, possibly with adjustment for factors such as document length and word frequency (Baeza-Yates and Ribeiro-Neto 1999; Manning and Schütze 1999; Jurafsky and Martin 2009). This approach has the advantage of being robust, straightforward to compute and is useful for identifying pairs of documents describing closely related topics.

Providing good recommendations is not easy, as items in collections can be similar in different ways. For example, two documents in a collection could be considered to be similar if they discuss the same topic or are written in the same style. The ways in which items can be considered similar also varies between collections. In collections of academic publications, such as Google Scholar or PubMed, pairs of citations could be considered to be similar for several reasons including being written by the same authors, citing the same publications, describing the same type of scientific investigation (e.g. a clinical trial or a meta study) or having the same conclusions. In different collections other features may be more relevant for determining whether items are similar. Existing methods for identifying similar items within collections do not acknowledge that there are *different ways* in which items can be similar.

*Personalised Access to Cultural Heritage Spaces (PATHS)*<sup>3</sup> was a three year project on the development of exploratory search interfaces for cultural heritage collections, including Europeana (Agirre *et al.* 2013a), funded by the European Commission under the *Digital Libraries and Digital Preservation Program*. The goal of the PATHS project is to create a system that helps users navigate through vast collections of art, offering a personalized tour according to the items they prefer. This system offers suggestions to the user as they navigate, offering similar or related items in some way, marking a path to follow. These paths can be based on anything: the theme or artist, the type of artwork, similar periods or places, etc. Users can create their own paths or follow other predefined paths.

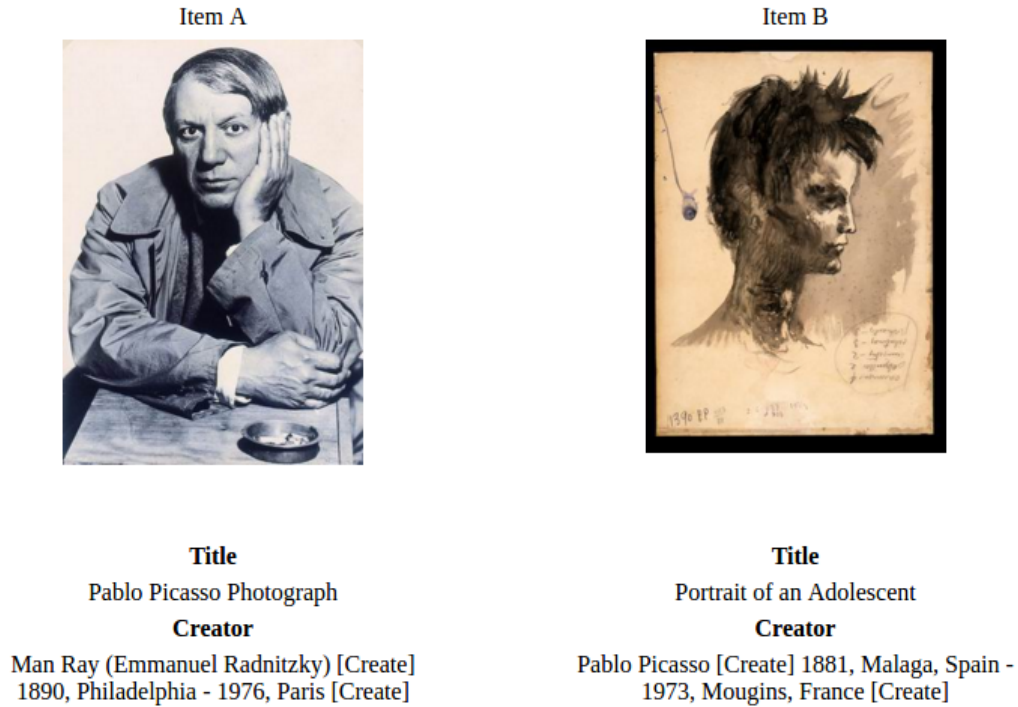
---

<sup>1</sup><http://scholar.google.com/>

<sup>2</sup><http://www.ncbi.nlm.nih.gov/pubmed>

<sup>3</sup><http://www.paths-project.eu>





**Figure 5.1** – Example of similar items, where the similarity is not based in the painting itself nor in the authors, but in the people involved in them.

The work described in this chapter is related to the PATHS project, and explores the problem of identifying different types of similarity<sup>4</sup> in a *large digital library* containing a collection of information about *cultural heritage* items, **Europeana** (see Section 5.2). The nature of the cultural heritage domain makes it appropriate for exploring the typed similarity problem. There are several ways in which the items in cultural heritage can be considered to be similar. For instance, in Figure 5.1 we can see two items, where person shown is not the same: the man on the left is '*Pablo Picasso*', the famous painter, and the boy in the right is an anonymous teenager. However, while the author of the photograph of '*Picasso*' is '*Emmanuel Radnitzky*', the author of the portrait of the teenager is '*Pablo Picasso*'. Although the items do not share the author either, we can see that '*Pi-*

<sup>4</sup>We use the term similarity since it is more commonly used in the research literature. We acknowledge the distinction between similarity and relatedness, and ask annotators to judge similarity between items (see later sections). However, the term similarity is used to capture both concepts for simplicity.

*casso*’ is involved in both, and with this information we can claim that these two items are similar because both of them are related to *Picasso*. Identifying these similarities has useful applications, including making recommendations (Resnick and Varian 1997; Grieser *et al.* 2007; Bohnert *et al.* 2009), supporting exploratory search (Marchionini 2006), personalisation (Bowen and Filippini-Fantoni 2004; O’Donnell *et al.* 2001) and (automatic) tour generation (Finkelstein *et al.* 2002; Roes *et al.* 2009; Agirre *et al.* 2013a).

This chapter also presents the first dataset for the typed similarity task. The dataset contains pairs of *Cultural Heritage* items from Europeana to which we assigned scores for a range of similarity types: similar author, similar people involved in the items, similar time period, similar location, similar event, similar subject and similar description. The dataset contains 1500 pairs of items that were manually annotated with those types using crowdsourcing. The annotators assigned a number between 0 (completely unrelated) to 5 (identical) for each type of similarity. The annotations are reliable, as demonstrated by high inter-tagger correlation agreement.

The next section describes Europeana, the digital library used in this study. Section 5.3 introduces the types of similarity that we used in this work and the method to gather and annotate the pairs of items that comprise our dataset. Section 5.4 presents some discussion and analysis of the dataset. Finally, Section 5.6 presents some conclusions.

## 5.2 Europeana

Europeana<sup>5</sup> is a web-portal that acts as a gateway to collections of cultural heritage items provided by a wide range of European institutions. It currently provides access to over 54 million digital records describing paintings, films, books, archival records and museum objects. The items are provided by around 1,500 institutions which range from major institutions, including the *Rijksmuseum* in Amsterdam, the *British Library* in London and the *Louvre* in Paris, to smaller and specialized organisations such as local museums. It therefore contains an aggregation of digital content from several sources and is not connected with any one physical museum.

Europeana stores the metadata about each item in an XLM-based format based on the Dublin Core standard. Information stored in this metadata includes a title (<dc:title>) and description (<dc:description>) for the item. There

---

<sup>5</sup><http://http://www.europeana.eu>

may also be information about the item's creator (e.g. painter, sculptor or photographer), stored in the `<dc:creator>` field, and date of creation, stored in the `<dc:date>` field. The date may be a specific date (e.g. 5th November 1905) or a time period (e.g. Bronze Age). The `<dc:collection>` field provides information about the collection the item came from (e.g. Kirklees Image Archive). Finally, cataloging information is provided for some items in the `<dc:subject>` field. This contains information about the item from a controlled vocabulary such as *Library of Congress Subject Headings*<sup>6</sup> or the *Art and Architecture Thesaurus*<sup>7</sup>. An example of metadata in the format used within Europeana is shown in Figure 5.2.

```
<dc:title>toy coins, crown (coin), toy coins</dc:title>
<dc:creator>The Fitzwilliam Museum, Cambridge, UK</dc:creator>
<dc:subject>Victoria (1837-1901) crown (coin) toy coins</dc:subject>
<dc:description>Artist: Victoria (1837-1901), ruler - Queen of
Great Britain 1837-1901; Date(s): 1887 - 1901; Classification(s):
toy coins, crown (coin), toy coins; Acquisition: given by Withers,
Paul, 2003-11-25 [CM.2666-2003]</dc:description>
```

**Figure 5.2** – Example of information about an item available in Europeana

The metadata are created by different content providers and vary significantly across items. Many of the items have only limited information associated with them, for example a very brief title. There is significant variation in the amount of information provided for some fields. For example, for some items the `<dc:description>` field contains over a thousand words of text while for others it is empty. In addition, the content providers that contribute to Europeana use different controlled vocabularies and it is not straightforward to establish correspondences between them. Some providers do not make any use of controlled vocabularies so there is no information in the `<dc:subject>` field for many items. This variation in the information available makes the problem of determining the similarity between items quite challenging.

---

<sup>6</sup><http://authorities.loc.gov/>

<sup>7</sup><http://www.getty.edu/research/tools/vocabularies/aat/>

## 5.3 A dataset for typed similarity

This section describes the construction of a manually annotated dataset for typed similarity generated from Europeana. First, we explain how we defined the different similarity types. Then we describe how the item pairs were selected. After that we describe in detail how we annotated the dataset and evaluate its quality. The dataset is freely available<sup>8</sup>.

### 5.3.1 Defining similarity types

The importance of typed-similarity was identified as part of PATHS<sup>9</sup>. The interface developed by the project provided information about similar items in collections and recommendations about items a user might like to consult. Users of the system requested more information about why items were considered similar. Consequently we explored methods for generating information about the type of similarity that could be presented to the user. Discussions with users and analysis of the collection revealed seven types of similarity:

1. **Similar author/creator** such as paintings by the same artist.
2. **Similar people involved** such as items showing the same people.
3. **Similar time period** such as items from the same year.
4. **Similar location** such as items showing the same place (e.g. a photograph and painting of the White House).
5. **Similar event or action involved** such as items showing weddings, or people eating ice cream.
6. **Similar subject** such as items related to the same subject, e.g. horses.
7. **Similar description** items which have a similar descriptions.

In addition, we also include a *general* similarity type where the annotators can express their overall impression of the similarity between both items.

---

<sup>8</sup><http://ixa2.si.ehu.es/sts>

<sup>9</sup><http://www.paths-project.eu>

### 5.3.2 Selecting item pairs

We have defined seven similarity types, and now we need to select items from Europeana that represent these types. Pairs of items were selected semi-automatically from Europeana. 25 pairs of items were manually selected for each of the seven similarity types (excluding general similarity), generating a total of 175 pairs. After removing duplicates and cleaning the dataset, 163 of these pairs remained. These manually selected pairs were then used as seeds to automatically select new pairs. The *Europeana API* was used to identify items that were similar to the seeds. For each seed, we created two chains of similar item pairs using an iterative process. The first chain of pairs was obtained using the current seed and a randomly chosen similar item from those provided by the Europeana API<sup>10</sup>. The newly identified item was then used as a new seed to continue building the chain of similar pairs. Thus, at each step, we obtained a new pair of similar items at *distance one*. The second chain followed the same iterative process, but selecting as new similar item among those appearing at *distance two* of the current seed in the chain. For each chain, we repeated the process up to five times.

This process yields 1,500 pairs, the 163 that were manually selected, 892 from *distance one* chains and 445 from *distance two* chains. We then divided the data into training and testing sets containing 750 pairs each. The training data contains 82 manually selected pairs, 446 pairs from *distance one* chains and 222 pairs with from *distance two* chains. The test data follows a similar distribution.

Table 5.1 shows descriptive statistics for the six fields provided to the participants (number of non-empty fields, average length of field in tokens and standard deviation of field length). These statistics were computed from the 1500 items (750 pairs) in the training portion of the dataset. A similar distribution was observed for the test set.

### 5.3.3 Annotation

After selecting the pairs as seen in the previous section, the dataset was annotated using *CrowdFlower*<sup>11</sup>, an online crowdsourcing platform. A survey was created containing the 1,500 pairs of the dataset (750 for training and 750 for testing). A set of 20 “gold” pairs with known answers were added for quality control<sup>12</sup>. Each annotator was initially shown four gold questions at the beginning for training, and then one gold question every two or four questions depending on the accuracy. If

<sup>10</sup>The Europeana API uses logs and textual descriptions to find similar items.

<sup>11</sup><http://www.crowdflower.com/>

<sup>12</sup>The gold pairs were chosen from those pairs manually selected by the authors.

Field	Non-empty	Avg. Length	Std. Dev.
Title	1500	5.9	4.5
Creator	1049	3.6	2.3
Subject	1434	7.8	7.4
Description	1469	77.0	169.4
Date	295	1.4	0.5
Source	21	1.3	0.9

**Table 5.1** – Corpus statistics for each of the fields in the training dataset.

the accuracy for a particular annotator dropped to less than 66.7% percent, the survey was stopped and the answers for that annotator discarded. Each annotator was allowed to rate a maximum of 20 pairs to avoid annotators becoming tired or bored. To ensure quality, the task was restricted to annotators from a set of English speaking countries: UK, USA, Australia, Canada and New Zealand. Each pair of items included eight questions regarding different types of similarity (see below) and was annotated at least by 5 annotators. A total of 1,584 annotators took part in the survey.

Figure 5.3 is a screenshot of the instructions provided to the annotators. Figure 5.4 shows how a pair of items from the dataset is presented to the annotators. Annotators were asked to rate the similarity between pairs of cultural heritage items in the range 0 to 5. A *Not Applicable* option was also included to avoid annotators being forced to make a choice when they were unsure. In those cases the similarity score was calculated using the values provided by the other annotators (or 0 if there were no other annotators for a particular item).

### 5.3.4 Quality of annotation

To assess annotation quality, we compute the Pearson product-moment correlation of each annotator against the average of the rest of the annotators, as in (Grieser *et al.* 2011; Aletras *et al.* 2012). We then averaged all the correlations. This measure is identical to the one used for evaluation (see Section 6.4) and can be used to put those results into context. The inter-tagger correlation in the dataset for each type of similarity shown in Table 5.2.

The correlation figures are high, with an average of 71.5, confirming that the task was well designed. The weakest correlations are for the *People Involved* and *Event or Action* types, suggesting they are the most difficult to identify. Other

## Estimate the Similarity between Cultural Heritage Items

### Instructions

[Hide](#)

The aim of this survey is to collect information about how people judge the relatedness of cultural heritage items in an online collection. You will be presented with pairs of cultural heritage items, including an image and additional textual information, and asked to judge how similar you think they are on the following scale:

- 5 - **Identical**
- 4 - **Strongly related**
- 3 - **Related**
- 2 - **Somewhat related**
- 1 - **Unrelated**
- 0 - **Completely unrelated**

For each pair you will be asked to provide a general similarity score, plus an additional score for each of the types of similarity considered, as follows:



- **Similar author**  
*(e.g. two items with the same creator should be rated 5 while two item with similar creators should be rated 4-3, etc)*
- **Similar people involved**  
*(e.g. two items showing the same people should be rated 5, two item showing children should be rated 4, showing similar people 4-3, etc)*
- **Similar time period**  
*(e.g. two items from 1941 should be rated 5, from the World War II should be rated 4, etc)*
- **Similar location**  
*(e.g. two items showing scenes of the same street should be rated 5, of London should be rated 4, etc)*
- **Similar event or action involved**  
*(e.g. two items showing weddings or people eating an ice-cream should be rated 5, etc)*
- **Similar subject**  
*(e.g. two items about cars or cats should be rated 5, etc)*
- **Similar description**  
*(e.g. two items with identical description should be rated 5, etc)*

Note that if you think that a particular similarity type is not relevant to a pair of items then you should select the "**Not Applicable**" choice. For example, this would be the correct option for the "**Author similarity**" if there is no information about the item's authors or creators.

**Figure 5.3** – Annotation instructions with explanations of each similarity type given to the annotators on CrowdFlower.



## CHAPTER 5. TYPED SIMILARITY

<p><b>Item 1</b></p>  <p><b>Title</b> Sculptured slabs of Aditya and Buddha, photographed at the Bihar Museum.</p> <p><b>Creator</b> Photographer : Beglar, Joseph David</p> <p><b>Subject</b> Bihar Bihar Sharif India Archaeological Survey of India Collections Archaeological Survey of India Collections (Indian Museum Series) Indian sculpture Indian sculpture (Buddhist) South Asia -- History 954</p> <p><b>Description</b> <span style="float: right;">I</span> This photograph showing sculpture fragments was taken by Joseph David Beglar in the 1870s. The sculptures were located in the Bihar museum and the photograph is part of the Archaeological Survey of India Collections. A note written by Bloch reads, "The sculptures photographed while exhibited in the Bihar Museum were collected from various places in Bihar, and are now in the Indian Museum.</p> <p><b>Date</b> <span style="float: right;">I</span> [1870]</p> <p><b>Source</b></p>	<p><b>Item 2</b></p>  <p><b>Title</b> Buddhist sculpture pieces from Jamal-Garhi. 1003995</p> <p><b>Creator</b> Photographer : Craddock, James</p> <p><b>Subject</b> North-West Frontier Province Pakistan Buddha images Gandharan art Indian sculpture Indian sculpture (Buddhist) museum objects South Asia -- History 954</p> <p><b>Description</b> Photograph of Buddhist sculpture pieces from Jamal-Garhi. This print shows boxed sculpture fragments. A note with Jamal-Garhi prints reads: 'The plates entered here also include photographs taken from sculptures coming from Takht-i-Bahl and Shahr-i-Buhlul. No separate arrangement was possible. Nearly all the sculptures coming from these places are now in the Indian Museum, Calcutta.'</p> <p><b>Date</b> [1880]</p> <p><b>Source</b></p>																		
<p><b>General Similarity (required)</b></p> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td style="width: 15%;"></td> <td style="width: 10%;"><b>0</b></td> <td style="width: 10%;"><b>1</b></td> <td style="width: 10%;"><b>2</b></td> <td style="width: 10%;"><b>3</b></td> <td style="width: 10%;"><b>4</b></td> <td style="width: 10%;"><b>5</b></td> <td style="width: 10%;"></td> </tr> <tr> <td>Completely Unrelated</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td>Identical</td> </tr> </table>			<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>		Completely Unrelated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Identical		
	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>													
Completely Unrelated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Identical												
<p><b>Author Similarity (required)</b></p> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td style="width: 15%;"></td> <td style="width: 10%;"><b>Not Applicable</b></td> <td style="width: 10%;"><b>0</b></td> <td style="width: 10%;"><b>1</b></td> <td style="width: 10%;"><b>2</b></td> <td style="width: 10%;"><b>3</b></td> <td style="width: 10%;"><b>4</b></td> <td style="width: 10%;"><b>5</b></td> <td style="width: 10%;"></td> </tr> <tr> <td>Completely Unrelated</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td>Identical</td> </tr> </table>			<b>Not Applicable</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>		Completely Unrelated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Identical
	<b>Not Applicable</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>												
Completely Unrelated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Identical											

**Figure 5.4** – Pair of items as shown in the survey to annotators. Only *general* and *author* similarity types are displayed here. The annotators would see all types.



### 5.3. A DATASET FOR TYPED SIMILARITY

---

Similarity type	Inter-tagger correlation
General	77.0%
Author	73.1%
People Involved	62.5%
Time Period	72.0%
Location	74.3%
Event or Action	63.9%
Subject	74.5%
Description	74.9%

**Table 5.2** – Inter-tagger correlation scores for each type of similarity on the Typed Similarity dataset.

annotations exercises which use a similar method to gather similarity annotations report comparable figures for inter-tagger agreement (Agirre *et al.* 2012).

We also computed confusion matrices for each of the similarity types (see Figure 5.5). The *General*, *Subject* and *Description* similarity fields (Figures 5.5a, 5.5g and 5.5h) show most of the weight in the 0-0 and 5-5 cells, indicating that there is a lot of agreement between annotators when they judge pairs as 0 or as 5. Almost all the disagreement is on 4-5 and 5-4 cells (i.e. very close disagreement).

The pattern is slightly different for the other similarity types (Figures 5.5b, 5.5c, 5.5d, 5.5e and 5.5f). In addition to the weight in the 0-0 and 5-5 cells there is also a lot of weight in the 0-5 and 5-0 cells. To discover the reason for this we manually examined a subset of the 0-5 and 5-0 disagreements. We found that they were mainly caused by one of the annotators ignoring the information in the description. A typical case would be two items with the same author where one of the items did not have a `dc:creator` field, but which mentioned who the author was in the description. The annotator who ignored the text in the description would assign a pair 0, while the annotator who had read the description would assign it a 5. Other than that we can conclude that annotators agree most of the time. As in the previous case, the fine-grained disagreement is also concentrated on the 4-5 and 5-4 cells for these similarity types.

Figures 5.6, 5.7 and 5.8 show the average score value distribution, as assigned by the annotators, separated into five ranges. The majority of pairs are very closely related with nearly half of the pairs in the [4-5] range. (The *Event* and *People Involved* similarity types are exceptions which exhibit smoother distributions.) The dataset is skewed towards higher similarity scores since our aim was to select

## CHAPTER 5. TYPED SIMILARITY

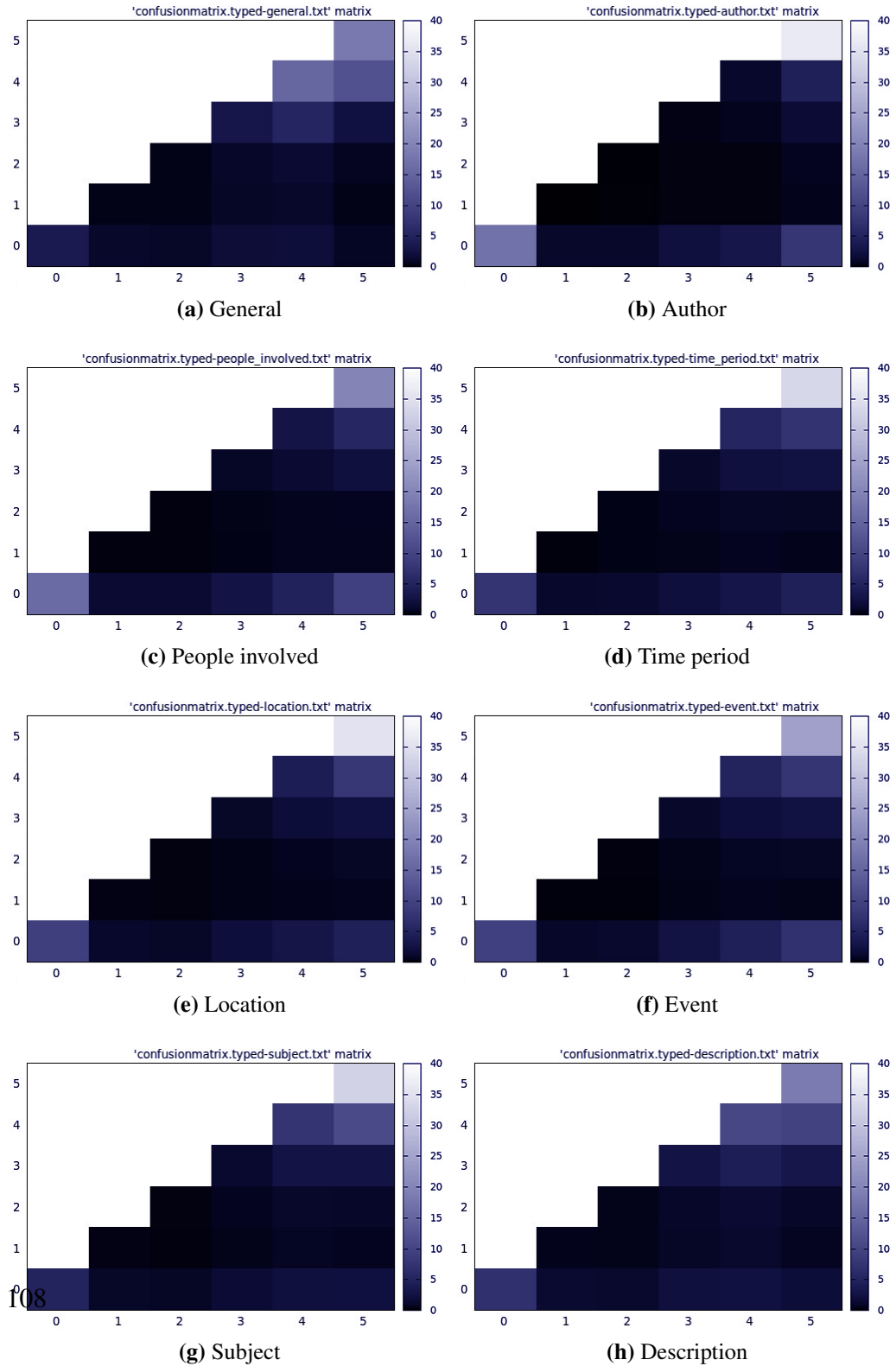
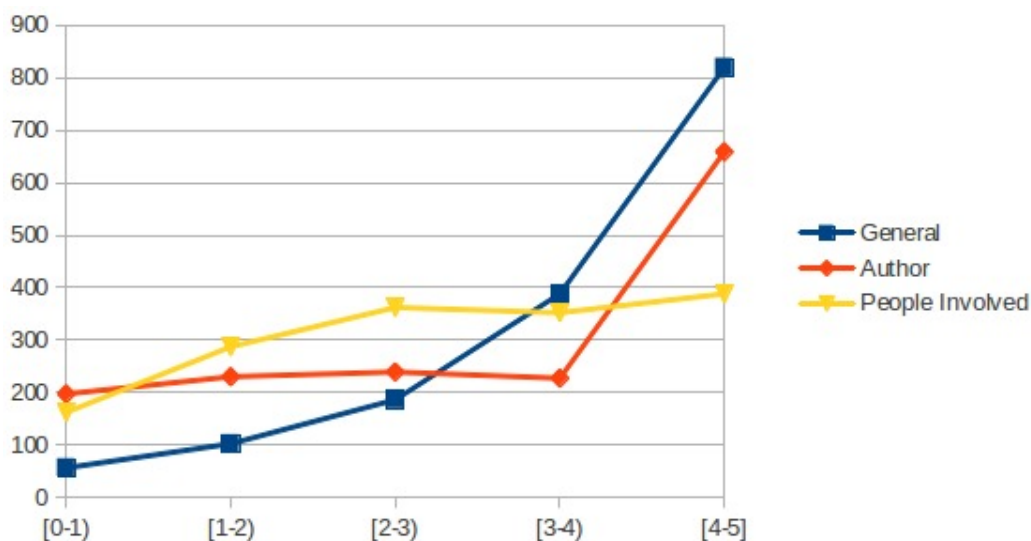


Figure 5.5 – Confusion matrices for the eight similarity types.



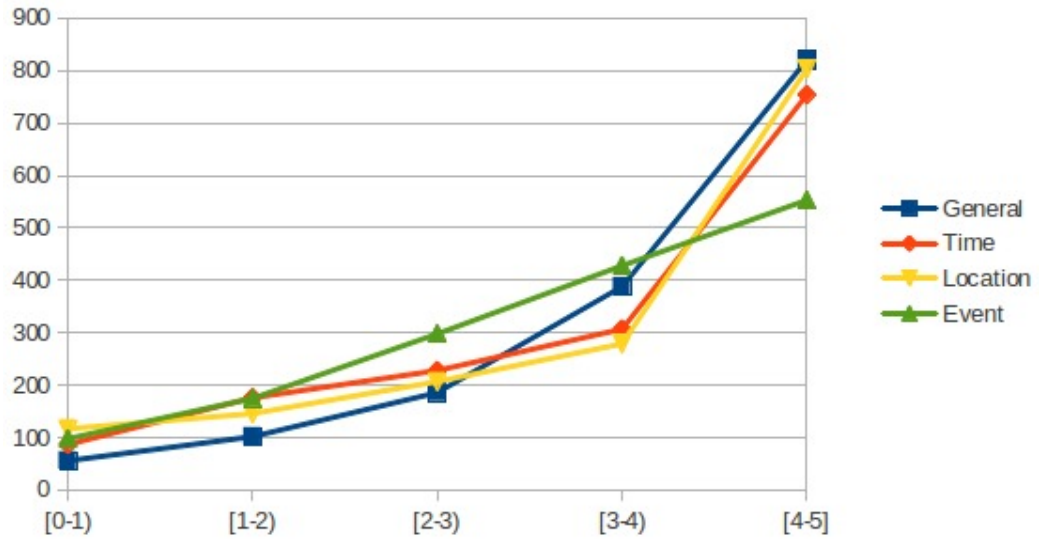
**Figure 5.6** – Score value distribution, as assigned by annotators, for general, author and people fields.

similar pairs of items rather than dissimilar ones.<sup>13</sup>

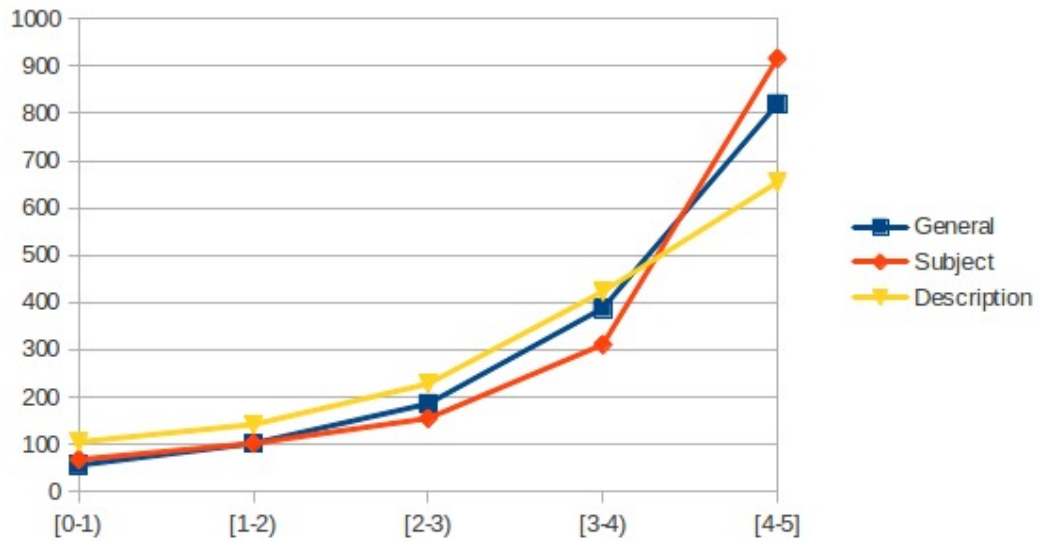
## 5.4 Discussion and analysis

We carried out an analysis of the annotations focussing on those pairs of items and annotation types where the annotators disagreed most. For instance, in the case of photographs (which form a substantial subset of the collection), there appears to be some confusion about the target of the annotation, specifically in relation to the *Author* similarity type. In these cases it is not clear whether the author type refers to the photographer who took the photograph or the creator of the item shown in the photograph (monument, building, painting, etc.). The same thing also happened for other types like *People Involved* in photographic items. Figure 5.9 shows an example of a pair of items where it is not clear if the annotation refers to the object in the picture or to the photograph itself. The title fields refer to sculptures of Buddha, but the creator fields refer to the photographers. Descriptions provide more information, but they also contribute to improve the

<sup>13</sup>Pearson is known to have issues when distributions are skewed. We checked the inter-tagger correlations using a down-sampled version of the full data, and the inter-tagger correlations we obtained were slightly higher.



**Figure 5.7** – Score value distribution, as assigned by annotators, for general, time period, location and event fields.



**Figure 5.8** – Score value distribution, as assigned by annotators, for general, subject and description fields.



uncertainty on what we are evaluating, as they describe both sculptures and photographs.

Another source of disagreement is the poor quality of metadata. For instance, the CREATOR field might contain the institution that keeps the item (e.g. *Fitzwilliam Museum*), a generic term (e.g. *staff*) or even just *none*. Some annotators assign the maximum score to cases where the term is the same for both items, while others read the description of the items, which specifies the author or indicates that it is unknown, and score the pair accordingly. Figure 5.10 shows a pair of items where the metadata indicates *staff* as creators and the description contains the actual author (designer) of both items (*Sir Bernard de Gomme*). One of the annotators, seeing that the metadata was not useful, rated the author similarity as *Not Applicable* (NA), while the rest did read the description and rated the author similarity accordingly. Taking the average produced 4 as the final value in the gold standard.

In another example (Figure 5.11) we can see the metadata for a pair of items, photograph of (different) bridges. The creator field lists '*unknown*' in one case and the author of the photograph in another ('*Eric de Mare*' is a well-known British photographer), but the description explicitly mentions the builders of each bridge: '*John Rennie*' for one, and his two sons, '*George*' and '*John*', for the other. The scores provided by the annotators of the *Author* similarity is 2, 3, 3, 0 and 0. In this case, it seems that the last two annotators have not read the full description of the items, while the first three did recognise that the authors of both bridges are related but not the same.

In order to explore the effect of incorrect or incomplete metadata on the annotation process we studied annotators' behaviour while completing the task. We enrolled some PhD students and asked them to annotate some of the conflicting pairs. We directly observed the annotators as they completed the task and also interviewed them after they had completed it. The study showed that the order of the fields and questions effected the annotations. For instance, the annotators rated the *Author* similarity before the *Description* similarity. In the absence of metadata in the CREATOR field some of the students evaluated this similarity as 0, without checking the description. They later identified the *Author* in the DESCRIPTION field, but some tended not to alter the score that has already been assigned for *Author* similarity. This study suggests that annotators can be confused by incorrect or incomplete metadata. For any future annotation exercises it would make sense to control the order in which the metadata is presented to the annotators so that the DESCRIPTION field is presented early (just after the title) since it provides the most general description for the item in most cases.

CHAPTER 5. TYPED SIMILARITY

<p>Item A</p> 	<p>Item B</p> 	
<p><b>Title</b></p> <p>Sculptured slabs of Aditya and Buddha, photographed at the Bihar Museum.</p> <p><b>Creator</b></p> <p>Photographer : Beglar, Joseph David</p> <p><b>Subject</b></p> <p>Bihar Bihar Sharif India Archaeological Survey of India Collections Archaeological Survey of India Collections (Indian Museum Series) Indian sculpture Indian sculpture (Buddhist) South Asia -- History 954</p> <p><b>Description</b></p> <p>This photograph showing sculpture fragments was taken by Joseph David Beglar in the 1870s. The sculptures were located in the Bihar museum and the photograph is part of the Archaeological Survey of India Collections. A note written by Bloch reads, "The sculptures photographed while exhibited in the Bihar Museum were collected from various places in Bihar, and are now in the Indian Museum. A paper by Mr Broadley, dealing with this collection, was published in Journal of the Asiatic Society of Bengal, vol. XLI, part I, 1872, pp. 209-311." Aditya is depicted on the left slab whilst Buddha can be seen in a reclining position on the right sculpture. There are also two architectural sculptures shown in the photograph .</p> <p><b>Date</b></p> <p>[1870]</p>	<p><b>Similarity type</b></p> <p><b>General:</b></p> <p>2.2</p> <p><b>Author:</b></p> <p>1.6</p> <p><b>Subject:</b></p> <p>2.6</p> <p><b>Description:</b></p> <p>2</p> <p><b>Time period:</b></p> <p>2.8</p>	<p><b>Title</b></p> <p>Buddhist sculpture pieces from Jamal-Garhi. 1003995</p> <p><b>Creator</b></p> <p>Photographer : Craddock, James</p> <p><b>Subject</b></p> <p>North-West Frontier Province Pakistan Buddha images Gandharan art Indian sculpture Indian sculpture (Buddhist) museum objects South Asia -- History 954</p> <p><b>Description</b></p> <p>Photograph of Buddhist sculpture pieces from Jamal-Garhi. This print shows boxed sculpture fragments. A note with Jamal-Garhi prints reads: 'The plates entered here also include photographs taken from sculptures coming from Takht-i-Bahl and Shahr-i-Buhlul. No separate arrangement was possible. Nearly all the sculptures coming from these places are now in the Indian Museum, Calcutta.'</p> <p><b>Date</b></p> <p>[1880]</p>

**Figure 5.9** – Sample pair of items, where it is not clear whether the annotators need to refer to the items in the photographs, or to the photographs themselves. For each item, the contents of the fields in the metadata are shown. In the center of the figure, the gold standard scores for each of the types is given.

<u>Similarity type</u>		
<b>Title</b>	<b>General:</b>	<b>Title</b>
Tilbury Fort, Tilbury, Essex	3.6	The Royal Citadel, Plymouth, Devon
<b>Creator</b>	<b>Author:</b>	<b>Creator</b>
staff	4	staff
<b>Subject</b>	<b>Subject:</b>	<b>Subject</b>
Aerial View Fort	3.8	Aerial View Coastal Fort Military
<b>Description</b>	<b>Description:</b>	<b>Description</b>
Tilbury Fort was designed in 1670 by Charles II's chief engineer, Sir Bernard de Gomme, in response to Dutch raids on the Thames. It is one of the best surviving examples of continental-inspired bastion defence in Britain.	3.4	Built between 1665-1667 on the site of Plymouth Fort, the Royal Citadel was designed by Dutch military engineer Bernard de Gomme to protect Plymouth from an attack by his own countrymen. More than three centuries later, the Citadel continues its military traditions and is now home to the British Army's 29 Commando Regiment.
<b>Date</b>	<b>Time period:</b>	<b>Date</b>
[1993]	4.2	[1999]
<b>Source</b>	<b>Location:</b>	<b>Source</b>
	3.4	

**Figure 5.10** – Sample of a pair of items which contain poor metadata in the author field (images removed for space). For each item, the contents of the fields in the metadata are shown. In the center of the figure, the gold standard scores for each of the types is given.

Overall our analysis suggests that although the quality of the annotation is very good, it may also be possible to improve it further. For instance, clarifying the photograph vs. item issue for the *Author* type and by providing specific instructions in face of poor quality metadata in order to pay more attention to the text in the description.

## 5.5 Systems evaluation

This Section describes the results of typed similarity systems that participated in the task. It presents the evaluation metrics used and the final results obtained using

## CHAPTER 5. TYPED SIMILARITY

---

	<u>Similarity type</u>	
<b>Title</b>	<b>General:</b>	<b>Title</b>
London Bridge, City of London	3.2	Serpentine Bridge, Hyde Park, Westminster, Greater London
<b>Creator</b>	<b>Author:</b>	<b>Creator</b>
not known	1.6	de Mare, Eric
<b>Subject</b>	<b>Subject:</b>	<b>Subject</b>
	3	Waterscape Animals Bridge Gardens And Parks
<b>Description</b>	<b>Description:</b>	<b>Description</b>
A view of London Bridge which is packed with horse-drawn traffic and pedestrians. This bridge replaced the earlier medieval bridge upstream. It was built by John Rennie in 1823-31. A new bridge, built in the late 1960s now stands on this site today.	2.2	The Serpentine Bridge in Hyde Park seen from the bank. It was built by George and John Rennie, the sons of the great architect John Rennie, in 1825-8.
<b>Date</b>	<b>Time period:</b>	<b>Date</b>
	4	[1945, 1980]
<b>Source</b>	<b>Location:</b>	<b>Source</b>
	3.8	

**Figure 5.11** – Sample of a pair of items which contain contradictory authorship information (images removed for space). For each item, the contents of the fields in the metadata are shown. In the center of the figure, the gold standard scores for each of the types is given.

the test data.

### 5.5.1 Evaluation metrics

System performance is evaluated by computing the Pearson product-moment correlation between the scores returned by the systems and the gold standard values (Rubenstein and Goodenough 1965), an approach often employed in word similarity experiments. This correlation can be obtained for each type of similarity, and the *mean correlation* score is the mean of the individual correlations.



### 5.5.2 The baseline system

The scores were produced using **TF-IDF-based similarity** (see Section 6.2.1) to provide an indication of the performance that could be obtained using a simple approach. This baseline system is described with more details in Section 6.3.2.

### 5.5.3 Results

This section describes the best *Typed Similarity* systems that participated in the competition. Table 5.3 shows the results the baselines and for the best seven runs. Each result is ordered by the rank of the system according to the mean of Pearson correlations on each similarity type.

Team and run	General	Author	People	Time	Location	Event	Subject	Description	Mean	Rank
Unitor-SVR_rbf	<b>.798</b>	<b>.816</b>	<b>.692</b>	<b>.747</b>	<b>.772</b>	<b>.684</b>	<b>.788</b>	<b>.800</b>	<b>.762</b>	1
Unitor-SVR_lin	.756	.808	.676	.709	.735	.662	.752	.775	.734	2
UBC_UOS-3*	.746	.666	.654	.741	.726	.655	.742	.776	.713	3
UBC_UOS-2	.746	.662	.652	.747	.724	.653	.740	.775	.712	4
ECNUCS-1	.604	.736	.366	.469	.384	.406	.523	.603	.511	5
UBC_UOS-1*	.726	.457	.447	.576	.486	.309	.502	.581	.510	6
ECNUCS-2	.606	.568	.366	.469	.384	.406	.556	.603	.495	7
baseline*	.669	.428	.446	.500	.484	.306	.502	.581	.489	8

**Table 5.3** – Test results of participants on the 2013 \*SEM shared task for each type of similarity, including the mean, and the rank of each run according to the mean. The systems marked with '\*' are described in the next chapter: our basic system is the baseline run, our improved system is UBC\_UOS-RUN1 and our machine learning system is UBC\_UOS-RUN3. UBC\_UOS-RUN2 was a variation of the ML system with a manual selection of features.

The best system among participant was the one presented by (Croce *et al.* 2013), which uses an approach to combine *Lexical Overlap* (LO) scores with *Distributional Compositional Semantics* (DCS) using *Support Vector Regression* (SVR). Their first step is to select specific phrases from the items, based on linguistic policies: word of specific *Part-of-Speech* (PoS), *Named Entities* (NE), mentions to specific name classes, such as *person*, *location*, or *date*. Summarizing, their selection is the following:

- **General:** *nouns, verbs and adjectives* plus *person, date and location* entities from from *dc>Title, dc:Subject* and *dc:Description* fields, and all tokens from *dc:Creator, dc:Date* and *dc:Source* fields.
- **Author:** all tokens from *dc:Creator* field and *person* entities from *dc:Description* field.
- **People involved:** *person* entities from *dc>Title, dc:Subject* and *dc:Description* fields.
- **Time period:** all tokens from *dc:Date* field and *date* entities from *dc>Title, dc:Subject* and *dc:Description* fields.
- **Location:** *location* entities from *dc>Title, dc:Subject* and *dc:Description* fields.
- **Event:** *nouns and verbs* from *dc>Title, dc:Subject* and *dc:Description* fields.
- **Subject:** *nouns, verbs and adjectives* from *dc>Title* and *dc:Subject* fields.
- **Description:** *nouns, verbs and adjectives* from *dc:Description* field.

To extract LO scores they lemmatize the selected phrases and then compute the *Jaccard similarity* between the words in both sentences ( $W_{S1}$  and  $W_{S2}$ ) using the following formula:

$$LO = \frac{|W_{S1} \cap W_{S2}|}{|W_{S1} \cup W_{S2}|} \quad (5.1)$$

Additional DCS scores are obtained by accounting the syntactic composition of the lexical information involved in the sentences. A basic lexical information is obtained using co-occurrence values from a Word Space following (Sahlgren 2006). Each phrase is represented using an *additive linear combination* or SUM operator. Then, the final score for each phrase is computed using cosine similarity between the vectors.

A second function is obtained by applying a DCS operator following the approach described in (Croce *et al.* 2012). Using a parser they generate dependency trees for the sentences and different triples  $(w_1, w_2, r)$  are generated, where  $w_1$  is the governor relation,  $w_2$  the dependent relation and  $r$  the grammatical type. Inspired by (Jimenez *et al.* 2012) they compute the *Soft Cardinality* with the following formula:

$$|S|'_{sim} \simeq \sum_{t_i}^{|T|} \left( \sum_{t_j}^{|T|} sim(t_i, t_j)^p \right)^{-1} \quad (5.2)$$

where  $T = \{t_1, \dots, t_n\}$  is a triple set,  $sim(t_i, t_j)$  is a similarity score between triples and  $p$  is a parameter to control the *softness* of the cardinality. Higher *Soft Cardinality* values mean that the elements are different, and lower values mean that elements are very similar. Given triples sets  $A$  and  $B$  they estimate their final score using a *Syntactic Soft Cardinality* (SSC) defined as follows:

$$SSC(A, B) = \frac{2|A \cap B|}{|A| + |B|} \quad (5.3)$$

Finally, they computed a *convolutional kernel-based similarity* score using a *Smoothed Partial Tree Kernel* (SPTK) as proposed in (Croce *et al.* 2011). SPTK allow to measure the similarity between syntactic tree structures.

To combine all these scores they used SVR using polynomial and *Radial Basis Function* (RBF) kernels, and their best run was the one using RBF kernels.

## 5.6 Conclusions

This chapter introduced the new problem of typed similarity, determining the type of the relation that holds between pairs of similar items. Typed similarity has various applications including providing recommendations and improving exploration through collections.

The problem was investigated within a subset of a large digital library of cultural heritage items from Europeana in the framework of the PATHS project. Seven types of similarity specific to this domain were identified: author, time, location, involved people, events, subject and description. A dataset was created using 1,500 pairs of items and annotated using crowdsourcing. It has been found that some types of similarity are more difficult to identify than others, such as the *People Involved* or *Events or Actions* represented in the items. In other cases, such as with the *Author*, there is also some confusion when dealing with photographs or paintings. However, an analysis of the annotation revealed an average Pearson correlation of 71.5, this high inter-annotator agreement indicates that the task is well-defined.

This work was done with the collaboration of Nikolaos Aletras and Mark Stevenson, from the University of Sheffield, and have led to a publication in a

## CHAPTER 5. TYPED SIMILARITY

---

journal.

In the next section we introduce several approaches to construct system that can be able to automatically identify the types of similarity defined in this chapter. We hope that these system could be used in Europeana to recommend similar items (and paths) to the users.

## A System for Typed Similarity

In this section we introduce several approaches to automatically identify the type of similarity, that has been used in a real-world application. We first explain how the text in the items was processed, followed by descriptions of the three systems we implemented, a baseline approach, a knowledge-based approach and a machine learning system. The high results obtained by our systems suggests that this technology is close to practical applications.

The work presented in this and the previous chapter have led to a publication in a journal.

### 6.1 Introduction

In the previous chapter we have presented a dataset for *Typed Similarity*. This datasets contains 1500 pairs of *Cultural Heritage* items, divided in 750 pairs for train and 750 for test. We found that the *Second Joint Conference on Lexical Computational Semantics* (\*SEM 2013) was the ideal framework to present the Typed Similarity dataset.

STS was selected as the official shared task of the \*SEM 2013 conference (Agirre *et al.* 2013c). We decided to take advantage of this opportunity to hold a pilot on Typed Similarity as a sub-task of STS, where the dataset was used to support a community evaluation exercise. The main objective of the task is to characterize the reason of **why** some item/element is similar to other. While STS reduces the problem of judging similarity to a single number, Typed Similarity measures the similarity by eight different numbers. For several applications it is

important to characterize why or how items are similar, that is why we add more distinct similarity scores.

The Typed Similarity dataset comprises pairs of Cultural Heritage items from Europeana, a gateway to collections of cultural heritage items, including more than 54 millions of sculptures, paintings, books, films, and other museum objects that have been digitized throughout a wide range of European institutions. A typical Europeana items contains meta-data describing a cultural heritage item and (usually) an image of the item itself.

Participants in the task where asked to submit the output of their systems with the eight similarity values between the items, computed employing all the information provided by the items meta-data. In addition to general similarity, participants need to score the specific kinds of similarity, as the ones seen previously in the Figure 5.3.

To demonstrate that the task was feasible we decided to participate as well. We presented three systems, the first is a basic system that was used as the *official baseline* of the task. The second system was a more advanced system that uses knowledge bases to judge the different similarities. Finally, we submitted a third system using the features of the previous two systems to feed a *Machine Learning* (ML) system.

In the next section we describe the similarity metrics/methods we later used on our systems. Section 6.3 describes in detail the three systems we submitted to the task, including how we processed the items. Section 6.4 described the official evaluation metrics of the task, presents the results of our system (including an error analysis), and compares the best systems that participated. Finally, Section 6.5 draws some conclusions.

## 6.2 Similarity methods

In this section we present the methods used for computing similarity and to build the typed-similarity systems described in the next section. Those tools are *Bag-of-Words* similarity using TF-IDF (Section 6.2.1), LDA (Section 6.2.2), the *Wikipedia Link Vector Model* (Section 6.2.3) and random walks over WordNet and Wikipedia graphs (Section 6.2.4). Each of these methods provide a different technique that can be applied to compute the similarity between a pair of texts.

LDA and WLVM scores were computed by Nikolaos Aletras, from the University of Sheffield, but we describe them here for completeness.

### 6.2.1 TF-IDF

A common approach for computing similarity between texts is to represent the documents as a *Bag-Of-Words* (**BOW**). Each BOW is a vector consisting of the words contained in the document in which each dimension corresponds to a word and the weight is the frequency with which the word occurs within the document. The similarity between two documents can be computed as the cosine of the angle between their vectors. If two documents are identical the cosine value of their vectors is 1 while if they share no common terms the cosine value is 0.

This approach is usually improved by giving more weight to words which occur in few documents and less weight to common words which tend to occur in many documents (e.g. *the*). We used the *Inverse Document Frequency* (IDF) (Baeza-Yates and Ribeiro-Neto 1999) using counts from the Culture Grid collection<sup>1</sup> in order to weight words. Thus, the **TF-IDF** similarity between items  $a$  and  $b$  is defined as follows:

$$sim_{\text{tf-idf}}(a, b) = \tag{6.1}$$

$$\frac{\sum_{w \in a, b} \text{tf}_{w,a} \times \text{tf}_{w,b} \times \text{idf}_w^2}{\sqrt{\sum_{w \in a} (\text{tf}_{w,a} \times \text{idf}_w)^2} \times \sqrt{\sum_{w \in b} (\text{tf}_{w,b} \times \text{idf}_w)^2}} \tag{6.2}$$

where  $\text{tf}_{w,x}$  is the frequency of the term  $w$  in  $x \in \{a, b\}$  and  $\text{idf}_w$  is the inverted document frequency of the word  $w$ .

### 6.2.2 LDA

*Latent Dirichlet Allocation* (**LDA**) (Blei *et al.* 2003) is a statistical method that learns a set of latent variables, called topics, describing the contents of a document collection. Given a topic model, documents can be viewed as a set of probability distributions over topics,  $\theta$ . The distribution for an individual document  $i$  is denoted as  $\theta_i$ .

The similarity between a pair of texts is estimated by comparing their topic distributions (Aletras *et al.* 2012; Aletras and Stevenson 2012). This is achieved by considering each distribution as a vector (consisting of the topics corresponding to an item and its probability) then computing the cosine of the angle between

<sup>1</sup>Culture Grid (<http://www.culturegrid.org.uk/>) is the digital content provider service from the Collection Trust and forms part of Europeana. It contains information about over one million items.

them, i.e.

$$sim_{LDA}(a, b) = \frac{\vec{\theta}_a \cdot \vec{\theta}_b}{|\vec{\theta}_a| \times |\vec{\theta}_b|} \quad (6.3)$$

where  $\vec{\theta}_x$  is the vector created from the probability distribution generated by LDA for text  $x$ .

To implement this approach an LDA model consisting of 100 topics was trained using the *gensim* package<sup>2</sup> with hyperparameters  $(\alpha, \beta)$  were set to  $1/num\_of\_topics$ .

### 6.2.3 WLVM

An algorithm described by (Milne and Witten 2008) associates Wikipedia articles with a document using machine learning techniques. We make use of that method to represent each item as a set of Wikipedia articles. The similarity of two documents can be thus computed as a function of the similarity between the Wikipedia articles associated with each text. We measured the similarity between Wikipedia articles using the *Wikipedia Link Vector Model (WLVM)* (Milne 2007), which uses both the link structure and the article titles. Each link is weighted by the probability of its occurrence. Thus, the value of the weight  $w$  for a link  $x \rightarrow y$  between articles  $x$  and  $y$  is:

$$w(x \rightarrow y) = |x \rightarrow y| \times \log \left( \sum_{z=1}^t \frac{t}{z \rightarrow y} \right) \quad (6.4)$$

where  $t$  is the total number of articles in Wikipedia. The similarity of articles is compared by forming vectors of the articles which are linked from them and computing the cosine of their angle. For example the vectors of two articles  $x$  and  $y$  are:

$$x = (w(x \rightarrow l_1), w(x \rightarrow l_2), \dots, w(x \rightarrow l_n)) \quad (6.5)$$

$$y = (w(y \rightarrow l_1), w(y \rightarrow l_2), \dots, w(y \rightarrow l_n)) \quad (6.6)$$

where  $x$  and  $y$  are two Wikipedia articles and  $x \rightarrow l_i$  is a link from article  $x$  to article  $l_i$ .

The similarity between two documents can then be computed by performing

---

<sup>2</sup><http://pypi.python.org/pypi/gensim>



pairwise comparison between the corresponding articles using WLVM, selecting the highest similarity score for each, as follows:

$$sim(a, b) = \frac{1}{2} \left( \frac{\sum_{w_1 \in a} \arg \max_{w_2 \in b} WLVM(w_1, w_2)}{|a|} + \frac{\sum_{w_2 \in b} \arg \max_{w_1 \in a} WLVM(w_2, w_1)}{|b|} \right) \quad (6.7)$$

where  $a$  and  $b$  are two texts,  $|a|$  the number of Wikipedia articles in  $a$  and  $WLVM(w_1, w_2)$  is the WLVM similarity between articles  $w_1$  and  $w_2$ .

### 6.2.4 Random walks

Random walks have been successfully used to compute the similarity between words (Agirre *et al.* 2010) and we extended these techniques to compute similarity between documents. We used the semantic disambiguation and similarity algorithm UKB<sup>3</sup> (Agirre and Soroa 2009), which applies *personalized PageRank* on a graph generated from the English WordNet (Christiane Fellbaum 1998), or alternatively, from Wikipedia.

To compute similarity between two words using UKB, we first represent WordNet as a graph  $G = (V, E)$ : graph nodes represent WordNet concepts (synsets) and dictionary words; relations among synsets are represented by undirected edges; and dictionary words are linked to the synsets associated to them by directed edges. We used the graph provided by UKB package. We then compute the personalized PageRank over WordNet separately for each of the words, producing two vectors with the probability distribution over WordNet synsets. The similarity between the words can be computed as the cosine between the two probability distributions.

The similarity between two documents can be computed initializing the random walks using the words in the respective texts to obtain a vector of probability distribution over synsets, and computing the cosine.

In addition to WordNet, we also used the Wikipedia graph, where the nodes correspond to Wikipedia articles, and the edges to hyperlinks between articles. We used version 3.0 of WordNet and the publicly available dump of Wikipedia dated 25th of May of 2011.

<sup>3</sup><http://ixa2.si.ehu.es/ukb/>

## 6.3 Constructing systems

In this section we present the three systems that were submitted to the Typed Similarity subtask of \*SEM 2013. The first system is a basic system that was designed to be used as the *official baseline* of the task. The second system is a knowledge-based approach, substituting some of the eight similarity scores assigned by the basic system. The third system uses the features of the previous two systems to feed a ML system.

### 6.3.1 Processing text in the items

The text in metadata the items was pre-processed using *Stanford CoreNLP* (Finkel *et al.* 2005; Toutanova *et al.* 2003), including *tokenization*, *Part-of-Speech tagging*, *Named Entity Recognition and Classification* (NERC) and *date detection*. The NERC module is key, as it allowed as to detect people, locations, organizations and dates. We used this entities on different ways, depending on the similarity type we were judging.

### 6.3.2 Baseline system

We implemented a baseline system using only **TF-IDF-based similarity** (see Section 6.2.1) to provide an indication of the performance that could be obtained using a simple approach. TF-IDF was applied differently for each similarity type:

- **General**: cosine similarity of TF-IDF vectors created using tokens from *all* fields.
- **Author**: cosine similarity of TF-IDF vectors created using *dc:Creator* field.
- **People Involved, Time Period** and **Location**: cosine similarity of TF-IDF vectors created from *people*, *locations* and *date* expressions recognized by NERC in *all* fields. Figure 6.1 shows a sample of the people, locations and dates which were automatically detected in the metadata for the item in Figure 5.2.
- **Events**: cosine similarity of TF-IDF vectors constructed from *verbs* in *all* fields.
- **Subject** and **Description**: cosine similarity of TF-IDF vectors created from respective fields.

```

<entity netype="ORG" lemma="Fitzwilliam_Museum" field="dc:creator"/>
<entity netype="LOC" lemma="Cambridge" field="dc:creator"/>
<entity netype="LOC" lemma="UK" field="dc:creator"/>
<entity netype="LOC" lemma="Victoria" field="dc:subject"/>
<entity netype="DATE" lemma="1837-1901" field="dc:subject"/>
<entity netype="LOC" lemma="Victoria" field="dc:description"/>
<entity netype="DATE" lemma="1837-1901" field="dc:description"/>
<entity netype="LOC" lemma="Great_Britain" field="dc:description"/>
<entity netype="DATE" lemma="1837-1901" field="dc:description"/>
<entity netype="DATE" lemma="1887_-_1901" field="dc:description"/>
<entity netype="PER" lemma="Withers" field="dc:description"/>
<entity netype="PER" lemma="Paul" field="dc:description"/>
<entity netype="DATE" lemma="2003-11-25" field="dc:description"/>

```

**Figure 6.1** – Example of NER analysis on the item shown in Figure 5.2

### 6.3.3 Knowledge based approach

The second approach was built on the baseline to make use of information from Wikipedia and WordNet (Section 6.2.4). Rather than applying TF-IDF similarity to all fields, as the baseline system did, different processes were applied to each field:

- **Author:** similarity using random walks on Wikipedia for the *person* entities in the *dc:Creator* field.
- **People Involved:** similarity using random walks on Wikipedia for the *person* entities recognized by NERC in *all* fields.
- **Location:** similarity using random walks on Wikipedia for the *location* entities recognized by NERC in *all* fields.
- **Events:** similarity using random walks on WordNet for *event verbs* and *nouns* in *all* fields. A list of verbs and nouns that may denote events was derived using morphosemantic links in WordNet<sup>4</sup>.

<sup>4</sup><http://wordnetcode.princeton.edu/standoff-files/morphosemantic-links.xls>

Results on the training data showed that the coverage of random walks for the aforementioned fields was quite low (except for event similarity, where good performance was obtained). This was caused by the large number of cases where the Stanford parser did not find entities which were in Wikipedia. Consequently the scores returned by the random walks were combined with the TF-IDF similarity scores presented in Section 6.3.2 as follows: if UKB similarity returns a score then it is multiplied with the TF-IDF score, otherwise we return the square of the TF-IDF similarity score.

In addition, the general similarity was improved in two ways: lemmas were used instead of word forms and Wikipedia was used to compute IDF scores instead of the *Culture Grid* collection. (We found that using *Culture Grid* lead to some undesirable outcomes, e.g. the word 'coin' had a very low IDF because it occurs very frequently in the CultureGrid collection.)

Finally, a dedicated similarity measure for dates was devised, in order to model that, e.g. 1500 and 1550 are similar dates while 99 and 1999 are not. To measure the time similarity between a pair of items we first need to identify the time expressions contained in both items. We assume that the year of creation or the year denoting when the event referenced by an item took place are good indicators of temporal similarity. Information about years mentioned in each item's meta-data is extracted using the following pattern:  $[1|2][0-9]\{3\}$ . Using this approach, each item is represented as a set of numbers denoting the years extracted from the item.

Time similarity between two items is computed based on the similarity between their associated years. Similarity between two years is defined as:

$$sim_{year}(y_1, y_2) = \max\{0, 1 - |y_1 - y_2| * k\} \quad (6.8)$$

where  $k$  is a parameter to weight the difference between two years, e.g. for  $k = 0.1$  all items that have difference of 10 years or more are assigned a score of 0. We experimented with various values for  $K$  and obtained the best results for  $k = 0.1$ .

Finally, time similarity between items  $a$  and  $b$  is computed as the maximum of the pairwise similarity between their associated years:

$$sim_{time}(a, b) = \max_{\forall i \in a, \forall j \in b} \{0, sim_{year}(a_i, b_j)\} \quad (6.9)$$

The, we substituted the preliminary *Time* similarity score of the baseline system by the measure obtained using the method presented in this section.

### 6.3.4 Machine learning system

The systems described so far used dedicated similarity measures to model each similarity type separately. In some cases, we are able to provide more than one option for each type of similarity. The machine learning system takes each of those similarity measures as features and uses linear regression (from Weka (Hall *et al.* 2009)) to learn models that fit those features to the training data.

To evaluate the general similarity, in addition to the TF-IDF cosine similarity used in the previous two systems, we used further similarity scores as features for general similarity, including LDA (Section 6.2.2) and WLVM (Section 6.2.3). We also used random walks (Section 6.2.4) to generate a probability distribution over WordNet synsets for all of the words in each item. Similarity between two words is computed by creating vectors from these distributions and comparing them using the cosine of the angle between the two vectors. If a word does not appear in WordNet its similarity value to every other word is set to 0.

Then, the similarity between a pair of items is computed by performing pairwise comparison between the words they contain and selecting the highest similarity score. The approach is similar to the one used to identify the similarity between a pair of texts based on their WLVM scores described in Section 6.2.3.

## 6.4 Evaluation

This section describes the evaluation of the typed similarity systems described previously. It presents the results obtained during the train and development phase (using the training portion of the dataset) and the final results obtained using the test data. Results are compared against state of the art systems. Note that we follow the same partition of training and test data that was used in the \*SEM 2013 shared task (see Section 6.4.3) making the results directly comparable.

### 6.4.1 Train and Development

The training data was used to develop the systems and check performance. Results for the machine learning system were generated using 10-fold cross-validation.

Table 6.1 shows the results obtained using the baseline system and improved components from the knowledge based system for each of the similarity types, including the improvement over the baseline. The differences in this table (the row with heading ' $\Delta$  Baseline') are multiplied by 100, to appreciate them better. The results show that the use of Wikipedia counts when computing TF-IDF improve

the results of general similarity, and yield the best results overall, with 6 absolute points of improvement over the baseline.

Type	Feature	Results	$\Delta$ Baseline
General	Baseline	0.658	-
	LDA	0.680	+2.2
	TF-IDF <sub>Wiki</sub>	<b>0.727</b>	<b>+6.9</b>
	UKB <sub>Wiki</sub>	0.541	-11.7
	WLVM	0.561	-9.7
Author	Baseline	0.396	-
	UKB <sub>Wiki</sub>	0.272	-12.4
	Combined UKB <sub>Wiki</sub>	0.447	<b>+5.1</b>
People involved	Baseline	<b>0.474</b>	-
	UKB <sub>Wiki</sub>	0.297	-17.7
	Combined UKB <sub>Wiki</sub>	0.465	-0.9
Location	Baseline	0.472	-
	UKB <sub>Wiki</sub>	0.222	-25.0
	Combined UKB <sub>Wiki</sub>	<b>0.480</b>	<b>+0.8</b>
Time	Baseline	0.548	-
	Improved Time Measure	<b>0.588</b>	<b>+4.0</b>
Event	Baseline	0.264	-
	UKB <sub>WN</sub>	<b>0.285</b>	<b>+2.1</b>
	Combined UKB <sub>WN</sub>	0.283	+1.8
Subject	Baseline	0.498	-
Description	Baseline	0.539	-

**Table 6.1** – Development results on each similarity type for the Baseline approach (TF-IDF) and the improved components applied in the knowledge based approach (cf. Sections 6.3.2 and 6.3.3). The differences in this table (the row with heading ' $\Delta$  Baseline') are multiplied by 100, to appreciate them better.

The use of random walks over Wikipedia (UKB<sub>Wiki</sub>) leads to results that are worse than the baseline approach, unless both scores are combined. (The combined score was obtained by multiplying the individual scores. If one of the algorithms did not yield a score, we squared the score of the other algorithm.) When a combination is used results improve for *Author* and *Location*, but not for *People Involved*. The use of random walks over WordNet (UKB<sub>WN</sub>) for events does improve over the baseline, without need of combination.

The dedicated time similarity measure also improves the results over the baseline. Note that we did not experiment with any improvements for the subject and description fields given the strong results generated by the baseline system.

The results of the full systems on each individual type in the training data are shown on Table 6.2, together with the mean score across all types. The table shows that the Baseline system (**Baseline**) obtains the lowest results, with the knowledge based system (**Knowledge**) getting better results overall and for most types (except for *People Involved*). Using a *Linear regression* (**ML system**) improves the results considerably for all types, yielding a mean value of 73.9. Values over 65 are obtained for all types, a values that is usually taken to mean a strong association.

System	General	Author	People	Time	Location	Event	Subject	Description	Mean
Baseline	.658	.396	.474	.548	.472	.264	.498	.539	.481
Knowledge	.727	.447	.465	.588	.480	.285	.497	.539	.503
ML system	<b>.787</b>	<b>.694</b>	<b>.697</b>	<b>.765</b>	<b>.749</b>	<b>.655</b>	<b>.759</b>	<b>.807</b>	<b>.739</b>

**Table 6.2** – Development results of each system for each type of similarity, including the mean over all types.

## 6.4.2 Test

Table 6.3 shows the results of our systems in the test dataset. The results are very similar to those obtained on the training data, but in this case the **Knowledge based system** performs better or equal to the baseline system in all types. The **Machine Learning system** provides the best results by far for all types, with correlations over 65 in all cases. The difference between the knowledge based system and baseline is not statistically significant, but the difference between the Machine Learning and knowledge based systems is (p-value < 0.02).

The high correlations obtained by our machine leaning system suggest that deploying automatic systems for typed-similarity in real tasks is feasible. In fact, the correlations attained by our best system (see Table 6.3) are comparable to the inter-tagger correlations obtained during annotation (see Section 5.3.4).

System	General	Author	People	Time	Location	Event	Subject	Description	Mean	Rank
Baseline	.669	.428	.446	.500	.484	.306	.502	.581	.489	8
Knowledge	.726	.457	.447	.576	.486	.309	.502	.581	.510	6
ML system	<b>.746</b>	<b>.666</b>	<b>.654</b>	<b>.741</b>	<b>.726</b>	<b>.655</b>	<b>.742</b>	<b>.776</b>	<b>.713</b>	<b>3</b>

**Table 6.3** – Test results of each system for each type of similarity, including the mean over all types and the rank of the systems among all participants.

### Error Analysis

In this section we perform an analysis of errors of our best system, the ML system. We first check the absolute difference between the *Gold Standard* (GS) and the value assigned by our system for each type of similarity. Table 6.4 shows the total numbers of pairs in the test data with an absolute error higher than 4, 3 or 2.

Absolute error	General	Author	People	Time	Location	Event	Subject	Description
> 4	0	0	0	0	0	0	0	0
> 3	1	13	8	2	4	5	4	0
> 2	47	77	52	44	64	41	52	34

**Table 6.4** – Number of pairs with an absolute error,  $abs(GS - MLsystem)$ , higher than 4, 3 or 2, on the Typed Similarity Test set (750 pairs). There were no errors over 4. Most of the errors over 3 concentrate on *Author* and *People Involved*, but errors over 2 are spread across all types.

Hand inspection reveals that errors related to the **General** similarity occur mainly when there is very low information, or when one of the items has a description much longer than the other. In these cases what happens is that the images provide the missing information, usually being very similar between them. In the case of the **Author** similarity, the worst errors occur when there is no metadata in the field, and authorship is indicated in the description. Using a pattern that looks for people entities after the word 'author' or 'creator' would be useful to solve this issue. Another source is the parser, when it does not detect the author as a



person. Errors in the **People Involved** similarity occur when the parser identifies as people references to locations (places with the name of people). The annotators are able to recognize this fact, but the system does not. The **Time Period** similarity have problems when it is implicit in historical facts, like *'World War II'*. It also happens if 'nineteenth-century' (or similar references) are used, instead of '19th century'. With **Location** similarity the main issue is similar to what happens with general similarity, failing when the metadata have different lengths. If an item has more mentions to places than the other, the system fails, even if it is clear for the annotators that the location of both items is the same.

As for **Event** similarity, the errors occur when the main event is diluted with other secondary actions. For example, in the pair 17 of the test set, the main event of both items is a *'car accident'*:

- Publicity photograph showing a severely damaged **crashed car**, no doubt the result of drink driving; 'give way' sign in background
- 22nd February, 1967 **Car crash** on a one way street after a 70mph chase. Several cars can be seen with police and members of the public gathered round. The stolen car had been chased along the inner ring road and the driver was arrested in Wade Lane

However, the causes are different: in one, the accident is caused by a drunk driver, and in the other it is a consequence of someone being pursued by police after having stolen a car. In addition, as in previous examples, the longer definition dilutes the main event.

The errors in **Subject** similarity usually occur when both fields mean the same, but they are written in a different way, or when a field is empty and the annotators base their decision on the image. In the case of **Description** similarity, when descriptions are of very different length the system tends to score lower than the annotators. The opposite occurs when the descriptions are cryptic, difficult to understand, or give technical information about the item, as the following example, where the annotators assign a 0, but the system assigns a 2.82:

- Poster, London & North Eastern Railway, 'Cleethorpes' by Andrew Johnson, printed by Waterlow & Sons Ltd., London & Dunstable. 1930. With bathers on the beach with a donkey ride and sea in the background. Format: quad royal. Dimensions: 40 x 50 inches, 1016 x 1270 mm.
- BR(LMR) Poster: Go abroad to the Isle of Man. Port St. Mary. Boats in Bay. Houses in background, by artist Peter Collins, printed by Waterlow

and Sons Ltd., London and Dunstable. Format: quad royal. Dimensions: 40 x 50 inches, 1016 x 1270mm.

### 6.4.3 Comparison to the best system

The **Baseline system** was used as the overall task baseline against which all runs were compared. This baseline system actually outperformed many of the submitted systems for various similarity types and achieved an overall ranking of 8th out of the 14 submitted systems.

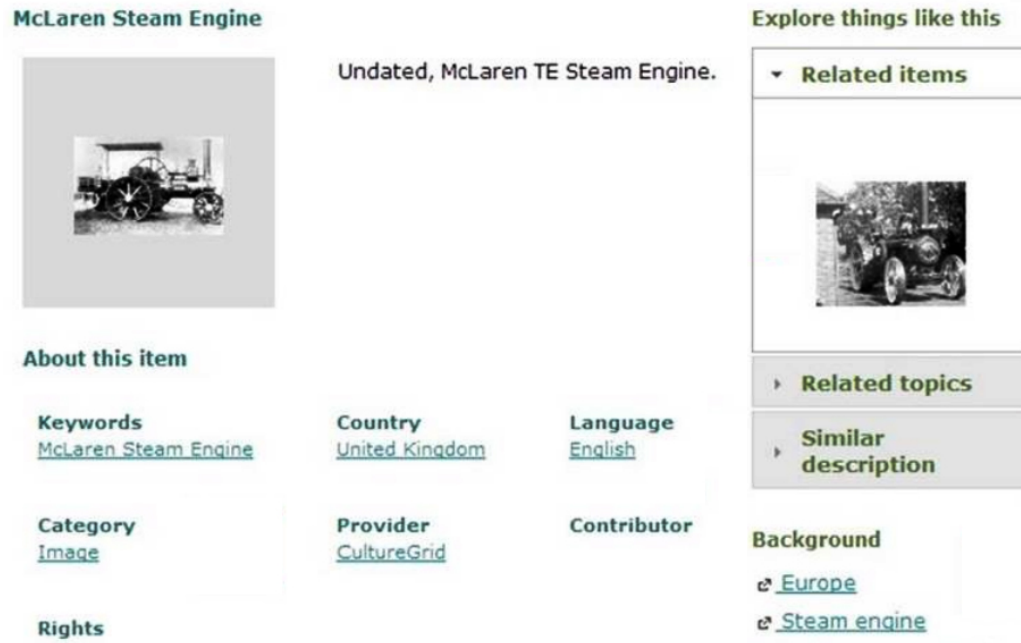
The **Knowledge based system** was ranked in 6th place overall and the **Machine Learning system** in 3rd place. Our systems achieved good correlation scores for almost all similarity types, with the exception of *Author* similarity, which is the worst ranked in comparison with the rest of the systems.

The best system (Croce *et al.* 2013) applied an approach that combined *Support Vector Regression* (SVR) with compositional distributional semantics to achieve an overall mean score of 0.762 across all similarity types. Full results are shown in Table 5.3.

## 6.5 Conclusion and Future Work

This chapter introduces three approaches to automatically determine the similarity type between cultural heritage items. The simplest approach was used as the official baseline of the *Typed Similarity* task, a community evaluation exercise within the \*SEM 2013 shared task on *Semantic Textual Similarity* (Agirre *et al.* 2013c). The exercise attracted 14 system runs from 6 teams. The baseline system was improved using knowledge-based and machine learning approaches. Our best results were obtained using the machine learning system which employed linear regression. This approach yields a mean Pearson correlation of 71.3, close to the human performance for this task.

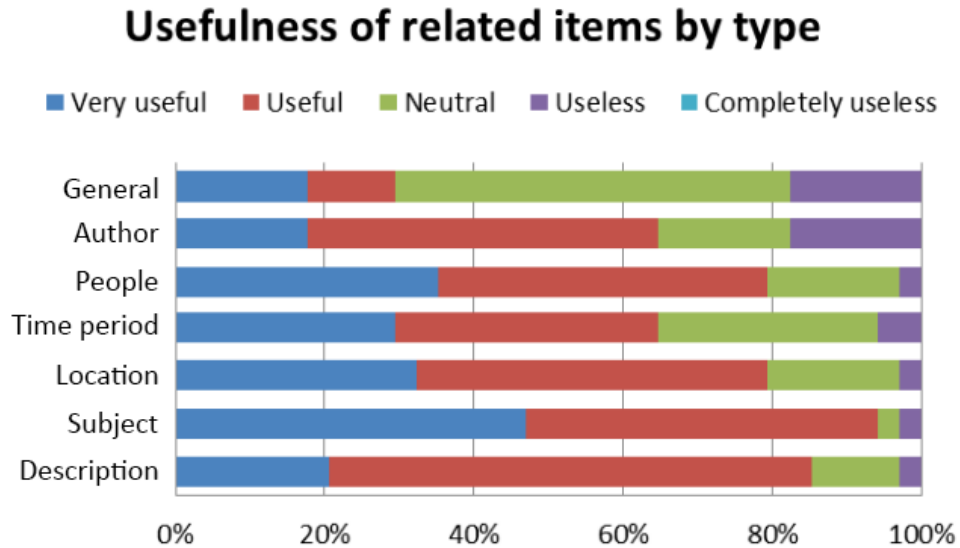
The typed similarity systems presented here have been deployed in PATHS within a prototype exploratory search interface for Europeana (Agirre *et al.* 2013a). When users view an individual Europeana item in this system they are also shown up to 25 similar items together with the similarity type to provide a motivation for displaying particular items. The type of the similarity is determined automatically using the machine learning system. A screenshot of this real-world application is shown in Figure 6.2, where for an item representing a 'McLaren Steam Engine' the system suggests three possible paths based on general, subject, and description similarity. We carried out further evaluation of this application to determine



**Figure 6.2** – Screenshot of the real recommender system, based on the system presented in this chapter, being used in Europaena to suggest paths to users.

how useful users find this information. A total of 88% of participant in this study responded that they found related items and path suggestions 'Very Useful' (35%) or 'Useful' (53%). We also asked participant to evaluate the usefulness of the different similarity types. Over 40% of participants responded that *similar Subject* was 'Very Useful', and over 85% responded that it was 'Very Useful' or 'Useful'. After *similar Subject*, the most popular similarity types were *People Involved*, *Location* and *Time Period*, in this order. *Author* similarity was the less popular for participant, although over 60% of them found it 'Very Useful' or 'Useful'. Full results of this study are shown in Figure 6.3.

These systems were constructed as a first approximation to the typed similarity problem. Although we obtained good results with them, we recognize that this system can be improved using more sophisticated techniques, such as the cube system presented in Chapter 4. Furthermore, for these system we only used the textual information contained in the metadata. More recent works have demonstrated that neural networks can be used to measure the similarity between images (Wang *et al.* 2014). These neural networks were not common when we released the Typed Similarity dataset, and the images were used only as a guidance for



**Figure 6.3** – Evaluation of the usefulness of the different similarity types. Similar *event* was left out of this study because it was not relevant enough in most items.

the annotators. It would be interesting to incorporate the knowledge provided by the thumbnails to the typed similarity systems.

In the future, we would like to explore the typed similarity problem in other domains, where a different set of similarity types are likely to be relevant.

This work was done with the collaboration of Nikolaos Aletras and Mark Stevenson, from the University of Sheffield, and has led to a publication in a journal.



## Conclusions and future work

This chapter presents a summary (Section 7.1) of the goals and contributions of this research on Semantic Textual Similarity. In Section 7.2 we list the publications that are related to this thesis. Finally, Section 7.3 proposes new lines of research.

### 7.1 Summary

Teaching computers to communicate through language is a real challenge. Language is full of phenomena that make comprehension very complex (e.g. polysemy, sarcasm, jokes, etc). In recent years there has been much progress in the field of NLP. In spite of this, we are still far from a complete *Natural Language Understanding* (NLU).

This thesis focuses on an aspect of NLU that attracts great interest, the evaluation of the *semantic similarity*. Evaluating whether two text fragments are similar to each other is a very important part in the field of *semantics* and NLP, and is useful for multiple tasks, such as *Machine Translation*, *Question Answering* or *Plagiarism detection*. Meaning equivalence measures can also be used to evaluate voice commands and understand the will of the user. This is specially useful for elder people, who has usually more difficulties using computers and other modern technologies, but also for home automation system to turn on the television or the home stereo system. Advanced systems could also understand that the command 'Buy me a flight to London or nearby' means 'Buy me a the cheapest plane ticket to London or to other surrounding airports'.

The main objectives of this thesis were to define tasks for the evaluation of semantic similarity, the creation of systems capable of assigning scores for se-

semantic similarity and creating datasets with which evaluate these systems. These three objectives have been completely fulfilled during the development of this thesis. We have defined two task for semantic similarity, *Semantic Textual Similarity* (STS) and *Typed Similarity*, and we have provided two highly competitive systems for each of the tasks. The datasets used to evaluate these systems were created in the scope of this research, and are widely used by other researchers, including to evaluate sentence representations generated using neural networks.

The first contribution of this work is to carry out a thorough review of the state of the art, starting with the concept of similarity in itself. We have reviewed the most common methods and techniques to compute semantic similarity, as well as the datasets that were available before the arrival of this thesis. Current systems and methods are data hungry, needing a lot of them to train their models. That is why the created datasets are of vital importance.

In this research we present STS and Typed Similarity, two tasks to evaluate the similarity between snippets of text and cultural heritage items, respectively. The first one aims to assess a graded similarity score between texts, while the second aims to explore different types of similarity, explaining how an item is similar to another. We have described the processes to define both tasks, and analysed the main problems that arose. We detail how the datasets were created, and how they were annotated, as well as all post-processes that helped to improve the quality of annotations. *Inter-tagger agreement* values demonstrate the high quality of datasets, and is the main reason of why they are so widely used by researchers all around the world. Semantic similarity has come a long way forward in this time: when the work presented in this thesis began, there were no datasets that would serve to train STS systems, it was not even clear how to carry out the task, nor how to evaluate the systems. Nowadays, there is great acceptance on the decisions taken in the definition of STS.

The work presented in this research includes the creation and annotation of 25 datasets for STS, which make a total of 15,436 pairs of sentences. This makes them the largest collection of data for STS. The quality of the annotations has been improved gradually each year, rising from an average inter-tagger of approximately 70% in 2012 to an approximately 83% in 2015. Created datasets are widely used, not only to evaluate STS systems, but also to evaluate the quality of *sentence representations* and other resources: several researchers are competing aiming to generate the best sentence representations or vectors, and they are using the datasets for STS to evaluate the quality of their *embeddings*. Recently, this area has become very popular, and there is great competition to achieve the best results on STS datasets using neural networks.

The dataset for Typed Similarity comprises 1,500 pairs of items from a Cultural Heritage collection, divided in 750 pairs for train and 750 for test. The annotation quality of the dataset is high, with an average of 71.5% Pearson correlation, confirming that the task was well designed. The weakest correlations are for the *People Involved* and *Event or Action* types, suggesting they are the most difficult to identify, but in any case, the lowest correlation are above 62.5%.

This thesis also presents two competitive systems, one for STS and other for Typed Similarity. The motivation behind the construction of the system for STS was to demonstrate that it is possible to use a new approach to combine knowledge-based methods with corpus-based methods without using *Machine Learning* (ML). Knowledge-based methods include knowledge from WordNet and Wikipedia, and corpus-based knowledge include two types of word vectors and three measures from clustering. Together with an heuristics metric for numbers, this information is used to construct a cube with all the knowledge, where each layer is a matrix with similarity scores from the different resources. An advantage of this system is that all knowledge is available at all times. No piece of information is discarded at any time, and is available for any algorithm until the final step, when the STS score is assigned. The system succeeds in extracting a result comparable to that obtained using ML on the sources of knowledge stored in the cube. Moreover, generating different features from the sentences and combining them with our system using ML makes it comparable to the best systems for STS. The system is also very robust and stable according to the ablation test performed, where resources were removed from the cube to measure their impact in the performance. It is important to note that this system does not combine with any other system to improve the results. For instance, it is very common to use the output of several state-of-the-art systems to feed a *Linear Regressor* or a *Support Vector Regressor*. Our system is an stand-alone system, which is not supported by another independent system. A preliminary version of this system was used in a system for *Interpretable STS* (Agirre *et al.* 2015b), and the similarity scores provided by this preliminary cube were the best among all submitted systems.

The objective for the Typed Similarity system was to demonstrate that the task was feasible. The system combines *TF-IDF similarity*, *Latent Dirichlet Allocation*, *Wikipedia Link Vector Model*, *Random Walks* on WordNet and Wikipedia and other heuristic metrics using ML. It works at a high level, close to *inter-tagger agreement* values, and thanks to its performance, it was possible to use it to recommend paths to users who visit an online museum (extensible to other similar platforms). A study on the usefulness of the similar items suggested by this systems showed that over 88% of the users that participated on the study evaluated

the system for Typed Similarity as *useful* or *very useful*. This system can be easily applied to other domains where it is useful to recommend products based on different types of similarity, such as online shopping platforms (e.g. Amazon, Ebay, Kiabi).

## 7.2 Publications

This section summarizes the main publications, organized by chapter, related to this thesis. Some of the publications have been made in collaboration with other people, and the authors are listed in alphabetical order. Unless explicitly indicated, the substantial part of the work presented in this thesis was done by me.

### Chapter 3:

<sup>1</sup>Agirre E., Banea C., Cer D.M., Diab M.T., Gonzalez-Agirre A., Mihalcea R., Rigau G., and Wiebe J. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In Bethard S., Cer D.M., Carpuat M., Jurgens D., Nakov P., and Zesch T., editors, *SemEval@NAACL-HLT*, 497–511. The Association for Computer Linguistics, 2016a. ISBN 978-1-941643-95-2

<sup>1</sup>Agirre E., Banea C., Cardie C., Cer D., Diab M., Gonzalez-Agirre A., Guo W., Lopez-Gazpio I., Maritxalar M., Mihalcea R., Rigau G., Uria L., and Wiebe J. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, CO, June 2015a. Association for Computational Linguistics

<sup>1</sup>Agirre E., Banea C., Cardie C., Cer D., Diab M., Gonzalez-Agirre A., Guo W., Mihalcea R., Rigau G., and Wiebe J. SemEval-2014 Task 10: Multilingual semantic textual similarity. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 81–91, Dublin, Ireland, August 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S14-2010>

<sup>1</sup>Agirre E., Cer D., Diab M., Gonzalez-Agirre A., and Guo W. \*SEM 2013 shared task: Semantic Textual Similarity. *Second Joint Conference on Lexical*

---

<sup>1</sup>Authors in alphabetical order



and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, 32–43, Atlanta, Georgia, USA, June 2013c. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S13-1004>

<sup>1</sup>Agirre E., Cer D., Diab M., and Gonzalez-Agirre A. Semeval-2012 task 6: A pilot on semantic textual similarity. \*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S12-1051>

#### Chapter 4:

Gonzalez-Agirre A., Agirre E., and Rigau G. Cubes for STS: Combining Knowledge-based and Corpus-based similarity. *TBS*, In preparation

<sup>1</sup>Agirre E., Gonzalez-Agirre A., Lopez-Gazpio I., Maritxalar M., Rigau G., and Uria L. UBC: Cubes for English Semantic Textual Similarity and Supervised Approaches for Interpretable STS. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015: Task 2)*, Denver, CO, June 2015b. Association for Computational Linguistics

#### Chapter 5:

Gonzalez-Agirre A., Aletras N., Rigau G., Stevenson M., and Agirre E. Why are these similar? Investigating item similarity types in a large Digital Library. *Journal of the Association for Information Science and Technology (JASIST)*, 67:7 pp. 1624-1638. John Wiley & sons. ISSN: 2330-1643. DOI: 10.1002/asi.23482, 2016

<sup>1</sup>Agirre E., Cer D., Diab M., Gonzalez-Agirre A., and Guo W. \*SEM 2013 shared task: Semantic Textual Similarity. *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, 32–43, Atlanta, Georgia, USA, June 2013c. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S13-1004>

### Chapter 6:

Gonzalez-Agirre A., Aletras N., Rigau G., Stevenson M., and Agirre E. Why are these similar? Investigating item similarity types in a large Digital Library. *Journal of the Association for Information Science and Technology (JASIST)*, 67:7 pp. 1624-1638. John Wiley & sons. ISSN: 2330-1643. DOI: 10.1002/asi.23482, 2016

<sup>1</sup>Agirre E., Aletras N., Gonzalez-Agirre A., Rigau G., and Stevenson M. UBC UOS-TYPED: Regression for typed-similarity. *The Second Joint Conference on Lexical and Computational Semantics (\*SEM 2013) pages 132-137 ISBN 978-1-937284-48-0*, 2013b

Other publications related to this thesis are listed below:

Lopez-Gazpio I., Maritxalar M., Gonzalez-Agirre A., Rigau G., Uria L., and Agirre E. Interpretable semantic textual similarity: Finding and explaining differences between sentences. *Knowl.-Based Syst.*, 119:186–199, 2017. URL <http://dx.doi.org/10.1016/j.knosys.2016.12.013>

<sup>1</sup>Agirre E., Gonzalez-Agirre A., Lopez-Gazpio I., Maritxalar M., Rigau G., and Uria L. Semeval-2016 task 2: Interpretable semantic textual similarity. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, 2016b

Gonzalez-Agirre A. and Rigau G. Construcción de una base de conocimiento léxico multilingüe de amplia cobertura: Multilingual Central Repository. *Revista para o Processamento Automático das Línguas Ibéricas (Linguamática)*. ISSN: 1647-0818. Vol. 5, Número 1. Pages: 13-28., 2013

Gonzalez-Agirre A., Laparra E., and Rigau G. Multilingual Central Repository version 3.0. *8th international conference on Language Resources and Evaluation (LREC'12) ISBN 978-2-9517408-7-7*, 2012c

Gonzalez-Agirre A., Laparra E., and Rigau G. Multilingual Central Repository version 3.0: upgrading a very large lexical knowledge base. *Proceedings of the 6th Global WordNet Conference (GWC'12) ISBN 978-80-263-0244-5.*, 2012d

Gonzalez-Agirre A., Castillo M., and Rigau G. A proposal for improving WordNet Domains. *8th international conference on Language Resources and Evaluation (LREC'12)* ISBN 978-2-9517408-7-7, 2012b

Gonzalez-Agirre A., Castillo M., and Rigau G. A graph-based method to improve WordNet Domains. *Proceedings of 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING'12)* ISBN 978-3-642-28603-2, 2012a

### 7.3 Future work

The system created for STS is not completely finished. Thanks to its design in the form of a cube with layers, we can always incorporate more knowledge, as new layers, but also in the way in which this knowledge is aggregated. Some of the resources have become obsolete, such as the Collobert and Weston word vectors. As new resources are created, it will be interesting introducing them into the system, either by replacing old versions or in conjunction with them. For example, *Glove* vectors (Pennington *et al.* 2014) could be a good addition to the cube layers. Moreover, during the design some layers were discarded, which could be incorporated as well. Solving the problems that arose when working with them is also an interesting direction. For example, we believe that a penalty/incompatibility layer could bring much knowledge to the cube.

Trying to extract more knowledge from the cube is also interesting. We are convinced that there is more knowledge than we have been able to extract. Finding optimal alignments within the cube is a motivating task. Brute force is not a solution to this problem, and one of the possibilities is to use *Interpretable STS* to train a system that knows how to find the right alignments. It will be necessary to create training sets that can help in this step, but first we must analyse the mental process humans carry out when doing this task. It is likely that more things have to be taken into account, such as looking at the *Part-of-Speech* of words we are aligning, or entity types, etc. Therefore, the next step will be to conduct an in deep analysis.

Another interesting direction is to incorporate the possibility of working with *compositionality* using the cube. Current system can be improved in this aspect, for instance when linking 'he surrendered' to 'gave himself up' and similar phrases such as 'Big Apple' and 'New York'. Although some of this *non-compositional* phenomena is captured by the system, many others are not cor-

rectly detected and aligned. Using sentence representations to detect this phenomena should improve our system. In the same way, we could use autoencoders to generate representations of phrases or intermediate nodes of dependency trees, extending the cube to use *word-phrase* and *phrase-phrase* similarities. This approach requires an aligning algorithm capable to find the best alignment for each pair of sentences and the computational cost is very high, as the algorithm should exclude the words and phrases that are already taking part in other alignments.

The Typed Similarity system was constructed as a first approximation, and even if it achieved good results, it can be improved using more sophisticated techniques. For instance, using the the cube system on the different similarity types should increase the performance. Furthermore, for these system we only used the textual information contained in the metadata, and recent works have demonstrated that neural networks can be used to measure the similarity between images ([Wang et al. 2014](#)). These neural networks were not common when we released the Typed Similarity dataset, and the images were used only as a guidance for the annotators. It would interesting to incorporate the knowledge provided by the thumbnails to the typed similarity system.

## Bibliography

- Agerri R. and Rigau G. Robust multilingual named entity recognition with shallow semi-supervised features. *Artificial Intelligence*, 238:63 – 82, 2016. ISSN 0004-3702. URL <http://www.sciencedirect.com/science/article/pii/S0004370216300613>.
- Agirre E., Aletras N., Clough P., Fernando S., Goodale P., Hall M., Soroa A., and Stevenson M. Paths: A system for accessing cultural heritage collections. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 151–156, Sofia, Bulgaria, August 2013a. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P13-4026>.
- Agirre E., Aletras N., Gonzalez-Agirre A., Rigau G., and Stevenson M. UBC UOS-TYPED: Regression for typed-similarity. *The Second Joint Conference on Lexical and Computational Semantics (\*SEM 2013) pages 132-137* ISBN 978-1-937284-48-0, 2013b.
- Agirre E. and Amigó E. Exploring evaluation measures for semantic textual similarity. *Unpublished manuscript*, In prep.
- Agirre E., Banea C., Cardie C., Cer D., Diab M., Gonzalez-Agirre A., Guo W., Lopez-Gazpio I., Maritxalar M., Mihalcea R., Rigau G., Uria L., and Wiebe J. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. *Proceedings of the 9th International Workshop on*

## BIBLIOGRAPHY

---

- Semantic Evaluation (SemEval 2015)*, Denver, CO, June 2015a. Association for Computational Linguistics.
- Agirre E., Banea C., Cardie C., Cer D., Diab M., Gonzalez-Agirre A., Guo W., Mihalcea R., Rigau G., and Wiebe J. SemEval-2014 Task 10: Multilingual semantic textual similarity. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 81–91, Dublin, Ireland, August 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S14-2010>.
- Agirre E., Banea C., Cer D.M., Diab M.T., Gonzalez-Agirre A., Mihalcea R., Rigau G., and Wiebe J. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In Bethard S., Cer D.M., Carpuat M., Jurgens D., Nakov P., and Zesch T., editors, *SemEval@NAACL-HLT*, 497–511. The Association for Computer Linguistics, 2016a. ISBN 978-1-941643-95-2.
- Agirre E., Cer D., Diab M., and Gonzalez-Agirre A. Semeval-2012 task 6: A pilot on semantic textual similarity. *\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S12-1051>.
- Agirre E., Cer D., Diab M., Gonzalez-Agirre A., and Guo W. \*SEM 2013 shared task: Semantic Textual Similarity. *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, 32–43, Atlanta, Georgia, USA, June 2013c. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S13-1004>.
- Agirre E., Cuadros M., Rigau G., and Soroa A. Exploring knowledge bases for similarity. *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC10). European Language Resources Association (ELRA). ISBN: 2-9517408-6-7. Pages 373–377.*", 2010.
- Agirre E., Gonzalez-Agirre A., Lopez-Gazpio I., Maritxalar M., Rigau G., and Uria L. UBC: Cubes for English Semantic Textual Similarity and Supervised Approaches for Interpretable STS. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015: Task 2)*, Denver, CO, June 2015b. Association for Computational Linguistics.

- Agirre E., Gonzalez-Agirre A., Lopez-Gazpio I., Maritxalar M., Rigau G., and Uria L. Semeval-2016 task 2: Interpretable semantic textual similarity. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, 2016b.
- Agirre E. and Soroa A. Personalizing pagerank for word sense disambiguation. *Proceedings of the 12th conference of the European chapter of the Association for Computational Linguistics (EACL-2009)*, Athens, Greece, 2009.
- Agirre E., Soroa A., Alfonseca E., Hall K., Kravalova J., and Pasca M. A study on similarity and relatedness using distributional and wordnet-based approaches. *Proceedings of annual meeting of the North American Chapter of the Association of Computational Linguistics*, 2009.
- Aletras N. and Stevenson M. Computing similarity between cultural heritage items using multimodal features. *Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, 85–93, Avignon, France, 2012.
- Aletras N., Stevenson M., and Clough P. Computing similarity between items in a digital library of cultural heritage. *J. Comput. Cult. Herit.*, 5(4):16:1–16:19, December 2012. ISSN 1556-4673.
- Allison L. and Dix T.I. A bit-string longest-common-subsequence algorithm. *Inf. Process. Lett.*, 23(6):305–310, December 1986. ISSN 0020-0190. URL <http://dl.acm.org/citation.cfm?id=8871.8877>.
- Arora S., Liang Y., and Ma T. A simple but tough-to-beat baseline for sentence embeddings. *International Conference on Learning Representations (ICLR 2017)*, 2017.
- Baeza-Yates R.A. and Ribeiro-Neto B.A. *Modern Information Retrieval*. ACM Press / Addison-Wesley, Essex, 1999. ISBN 0-201-39829-X.
- Baker C.F., Fillmore C.J., and Lowe J.B. The berkeley framenet project. *COLING '98 Proceedings of the 17th international conference on Computational linguistics - Volume 1*, 1998.
- Bar D., Biemann C., Gurevych I., and Zesch T. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. *Proceedings of the 6th International Workshop on Semantic Evaluation, in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, 2012.

## BIBLIOGRAPHY

---

- Baroni M., Dinu G., and Kruszewski G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2014)*, 238–247. Association for Computational Linguistics, 2014a. URL <http://aclweb.org/anthology/P14-1023>.
- Baroni M., Dinua G., and Kruszewski G. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, 1:238–247, 2014b. URL [https://www.researchgate.net/publication/270877599\\_Don%27t\\_count\\_predict\\_A\\_systematic\\_comparison\\_of\\_context-counting\\_vs\\_context-predicting\\_semantic\\_vectors](https://www.researchgate.net/publication/270877599_Don%27t_count_predict_A_systematic_comparison_of_context-counting_vs_context-predicting_semantic_vectors).
- Barrón-Cedeño A., Rosso P., Agirre E., and Labaka G. Plagiarism detection across distant language pairs. *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10*, 37–45, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=1873781.1873786>.
- Bengio Y., Ducharme R., Vincent P., and Janvin C. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=944919.944966>.
- Bentivogli L., Bernardi R., Marelli M., Menini S., Baroni M., and Zamparelli R. Sick through the semeval glasses. lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *Language Resources and Evaluation*, 50(1): 95–124, 2016. ISSN 1574-0218. URL <http://dx.doi.org/10.1007/s10579-015-9332-5>.
- Best C., van der Goot E., Blackler K., Garcia T., and Horby D. Europe media monitor - system description. *EUR Report 22173-En*, Ispra, Italy, 2005.
- Biemann C. Creating a system for lexical substitutions from scratch using crowdsourcing. *Lang. Resour. Eval.*, 47(1):97–122, March 2013. ISSN 1574-020X. URL <http://dx.doi.org/10.1007/s10579-012-9180-5>.



- Bird S. Nltk: the natural language toolkit. *Proceedings of the COLING/ACL on Interactive presentation sessions*, 69–72. Association for Computational Linguistics, 2006.
- Blei D.M., Ng A.Y., and Jordan M.I. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003. ISSN 1532-4435.
- Bohnert F., Schmidt D., and Zuckerman I. Spatial Process for Recommender Systems. *Proc. of IJCAI 2009, 2022–2027*, Pasadena, CA, 2009.
- Bowen J. and Filippini-Fantoni S. Personalization and the Web from a Museum Perspective. *Proc. of Museums and the Web 2004*, 63–78, 2004.
- Bowman S.R., Angeli G., Potts C., and Manning C.D. A large annotated corpus for learning natural language inference. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.
- Briscoe T. and Boguraev B. *Computational lexicography for natural language processing*. Longman Publishing Group, White Plains, NY, USA, 1989. ISBN 0-470-21187-3.
- Broder A. On the resemblance and containment of documents. *Proceedings of the Compression and Complexity of Sequences 1997*, SEQUENCES '97, 21–, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-8186-8132-2. URL <http://dl.acm.org/citation.cfm?id=829502.830043>.
- Brown P.F., deSouza P.V., Mercer R.L., Pietra V.J.D., and Lai J.C. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, December 1992. ISSN 0891-2017. URL <http://dl.acm.org/citation.cfm?id=176313.176316>.
- Callison-Burch C., Fordyce C.a., Koehn P., Monz C., and Schroeder J. (meta-) evaluation of machine translation. *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, 136–158, 2007. URL <http://dl.acm.org/citation.cfm?id=1626355.1626373>.
- Callison-Burch C., Fordyce C., Koehn P., Monz C., and Schroeder J. Further meta-evaluation of machine translation. *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, 70–106, 2008. ISBN 978-1-932432-09-1. URL <http://dl.acm.org/citation.cfm?id=1626394.1626403>.

## BIBLIOGRAPHY

---

- Chaves R.P. Wordnet and automated text summarization. *Proceedings of 6th Natural Language Processing Pacific Rim Symposium NLPRS 2001*, 109–116, Tokyo, Japan, Jan 2001.
- Chen D.L. and Dolan W.B. Collecting highly parallel data for paraphrase evaluation. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 190–200, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL <http://dl.acm.org/citation.cfm?id=2002472.2002497>.
- Cho K., van Merriënboer B., Gülçehre Ç., Bougares F., Schwenk H., and Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL <http://arxiv.org/abs/1406.1078>.
- Christiane Fellbaum C. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
- Chung J., Gülçehre Ç., Cho K., and Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL <http://arxiv.org/abs/1412.3555>.
- Church K.W. and Hanks P. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16:22–29, March 1990. ISSN 0891-2017. URL <http://portal.acm.org/citation.cfm?id=89086.89095>.
- Clark A. Combining distributional and morphological information for part of speech induction. *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, 59–66, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. ISBN 1-333-56789-0. URL <http://dx.doi.org/10.3115/1067807.1067817>.
- Clough P. and Stevenson M. Developing a corpus of plagiarised short answers. *Language Resources and Evaluation*, 45(1):5–24, 2011. ISSN 1574-020X.
- Collobert R. and Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, 160–167, New York, NY, USA, 2008. ISBN 978-1-60558-205-4.

- Croce D., Moschitti A., and Basili R. Structured lexical similarity via convolution kernels on dependency trees. *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, 1034–1046, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL <http://dl.acm.org/citation.cfm?id=2145432.2145544>.
- Croce D., Storch V., Annesi P., and Basili R. Distributional compositional semantics and text similarity. *Proceedings of the 2012 IEEE Sixth International Conference on Semantic Computing, ICSC '12*, 242–249, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4859-3. URL <http://dx.doi.org/10.1109/ICSC.2012.63>.
- Croce D., Storch V., and Basili R. Unitor-core\_typed: Combining text similarity and semantic filters through sv regression. *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, 59–65, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S13-1007>.
- Dagan I., Dolan B., Magnini B., and Roth D. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering*, 16:105–105, 2010. ISSN 1469-8110. URL [http://journals.cambridge.org/article\\_S1351324909990234](http://journals.cambridge.org/article_S1351324909990234).
- Dagan I., Glickman O., and Magnini B. The pascal recognising textual entailment challenge. *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment, MLCW'05*, 177–190, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33427-0, 978-3-540-33427-9. URL [http://dx.doi.org/10.1007/11736790\\_9](http://dx.doi.org/10.1007/11736790_9).
- David L. Chen D.L. and Dolan W.B.D. Collecting highly parallel data for paraphrase evaluation. *Proceedings of the 49th Annual Meetings of the Association for Computational Linguistics (ACL)*, 2011.
- Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K., and Harshman R. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

## BIBLIOGRAPHY

---

- Dennis S. The construction of a thesaurus automatically from a sample of text. *Statistical association methods for mechanized documentation, symposium proceedings (Miscellaneous publication 269)*. Washington, DC: National Bureau of Standards, 1964.
- Dinu L.P. and Popescu M. Ordinal measures in authorship identification. In *Proceedings of the 3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*, 2009.
- Dolan B., Quirk C., and Brockett C. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 350, 2004.
- Dolan W. and Brockett C. Automatically constructing a corpus of sentential paraphrases. *3rd International Workshop on Paraphrasing (IWP2005)*, 2005.
- Dreyer M. and Marcu D. Hyter: Meaning-equivalent semantics for translation evaluation. *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, 2012.
- Drucker H., Burges C.J.C., Kaufman L., Smola A.J., and Vapnik V. Support vector regression machines. In Mozer M.C., Jordan M.I., and Petsche T., editors, *Advances in Neural Information Processing Systems 9*, 155–161. MIT Press, 1997. URL <http://papers.nips.cc/paper/1238-support-vector-regression-machines.pdf>.
- Dunning T. Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 19:61–74, March 1993. ISSN 0891-2017. URL <http://portal.acm.org/citation.cfm?id=972450.972454>.
- Dzikovska M.O., Moore J.D., Steinhauser N., Campbell G., Farrow E., and Callaway C.B. Beetle II: a system for tutoring and computational linguistics experimentation. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-2010) demo session*, 13–18, Uppsala, Sweden, July 2010. URL <http://homepages.inf.ed.ac.uk/mdzikovs/papers/dzikovska-beetle-demo-acl2010.pdf>.
- E. Ukkonen E. Algorithms for approximate string matching. *Information and Contro*, 64:110–118, 1985.

- Erhan D., Bengio Y., Courville A., Manzagol P.A., Vincent P., and Bengio S. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.*, 11:625–660, March 2010. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=1756006.1756025>.
- Erkan G. and Radev D.R. LexRank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, 2004. ISSN 1076-9757. URL <http://dl.acm.org/citation.cfm?id=1622487.1622501>.
- Fellbaum C. *WordNet - An Electronic Lexical Database*. MIT Press, 1998.
- Finkel J.R., Grenager T., and Manning C. Incorporating non-local information into information extraction systems by gibbs sampling. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05*, 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL <http://dx.doi.org/10.3115/1219840.1219885>.
- Finkelstein L., Gabrilovich E., Matias Y., Rivlin E., Solan Z., Wolfman G., and Ruppin E. Placing Search in Context: The Concept Revisited. *ACM Transactions on Information Systems*, 20(1):116–131, 2002.
- Firth J.R. The technique of semantics. *Transactions of the Philological Society*, 34(1):36–73, 1935. ISSN 1467-968X. URL <http://dx.doi.org/10.1111/j.1467-968X.1935.tb01254.x>.
- Firth J. *Papers in linguistics, 1934-1951*. Oxford University Press, 1957. URL <https://books.google.es/books?id=jDu3AAAAIAAJ>.
- Gabrilovich E. and Markovitch S. Computing semantic relatedness using wikipedia-based explicit semantic analysis. *Proceedings of the 20th international joint conference on Artificial intelligence, IJCAI'07*, 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
- Ganitkevitch J., Van Durme B., and Callison-Burch C. PPDB: The paraphrase database. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, 758–764, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL <http://cs.jhu.edu/~ccb/publications/ppdb.pdf>.

## BIBLIOGRAPHY

---

- Goikoetxea J., Soroa A., and Agirre E. Random walks and neural network language models on knowledge bases. *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1434–1439, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N15-1165>.
- Gonzalez-Agirre A., Agirre E., and Rigau G. Cubes for STS: Combining Knowledge-based and Corpus-based similarity. *TBS*, In preparation.
- Gonzalez-Agirre A., Aletras N., Rigau G., Stevenson M., and Agirre E. Why are these similar? Investigating item similarity types in a large Digital Library. *Journal of the Association for Information Science and Technology (JASIST)*, 67:7 pp. 1624-1638. John Wiley & sons. ISSN: 2330-1643. DOI: 10.1002/asi.23482, 2016.
- Gonzalez-Agirre A., Castillo M., and Rigau G. A graph-based method to improve WordNet Domains. *Proceedings of 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING'12)* ISBN 978-3-642-28603-2, 2012a.
- Gonzalez-Agirre A., Castillo M., and Rigau G. A proposal for improving WordNet Domains. *8th international conference on Language Resources and Evaluation (LREC'12)* ISBN 978-2-9517408-7-7, 2012b.
- Gonzalez-Agirre A., Laparra E., and Rigau G. Multilingual Central Repository version 3.0. *8th international conference on Language Resources and Evaluation (LREC'12)* ISBN 978-2-9517408-7-7, 2012c.
- Gonzalez-Agirre A., Laparra E., and Rigau G. Multilingual Central Repository version 3.0: upgrading a very large lexical knowledge base. *Proceedings of the 6th Global WordNet Conference (GWC'12)* ISBN 978-80-263-0244-5., 2012d.
- Gonzalez-Agirre A. and Rigau G. Construcción de una base de conocimiento léxico multilingüe de amplia cobertura: Multilingual Central Repository. *Revista para o Processamento Automático das Línguas Ibéricas (Linguamática)*. ISSN: 1647-0818. Vol. 5, Número 1. Pages: 13-28., 2013.
- Grefenstette G. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Norwell, MA, USA, 1994. ISBN 0792394682.

- Grieser K., Baldwin T., and Bird S. Dynamic Path Prediction and Recommendation in a Museum Environment. *Proc. of the Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2007)*, 49–56, Prague, Czech Republic, 2007.
- Grieser K., Baldwin T., Bohnert F., and Sonenberg L. Using Ontological and Document Similarity to Estimate Museum Exhibit Relatedness. *Journal of Computing and Cultural Heritage (JOCCH)*, 3(3):1–20, 2011. ISSN 1556-4673.
- Guo W., Li H., Ji H., and Diab M. Linking tweets to news: A framework to enrich online short text data in social media. *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*, 239–249, 2013.
- Gusfield D. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, NY, USA, 1997. ISBN 0-521-58519-8.
- Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I.H. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, November 2009. ISSN 1931-0145. URL <http://doi.acm.org/10.1145/1656274.1656278>.
- Han L., Kashyap A.L., Finin T., Mayfield J., and Weese J. UMBC\_EBIQUITY-CORE: Semantic Textual Similarity Systems. *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, June 2013.
- Han L., Martineau J., Cheng D., and Thomas C. Samsung: Align-and-differentiate approach to semantic textual similarity. *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, 172–177, 2015. URL <http://aclweb.org/anthology/S/S15/S15-2031.pdf>.
- Harris Z.S. *Mathematical Structures of Language*. Wiley, New York, NY, USA, 1968.
- Hatzivassiloglou V., Klavans J.L., and Eskin E. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. *Proceedings of the 1999 joint sigdat conference on empirical methods in natural language processing and very large corpora*, 203–212. Citeseer, 1999.



## BIBLIOGRAPHY

---

- Hearst M. *Search User Interfaces*. Cambridge University Press, 2009.
- Hinton G.E., Osindero S., and Teh Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. ISSN 0899-7667. URL <http://dx.doi.org/10.1162/neco.2006.18.7.1527>.
- Hirst G. and St-Onge D. *WordNet: An Electronic Lexical Database - Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms*, in *Wordnet: An Electronic Lexical Database*, chapter 13, 305–332. MIT Press, 1998.
- Hodosh M., Young P., and Hockenmaier J. Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res.*, 47(1):853–899, May 2013. ISSN 1076-9757. URL <http://dl.acm.org/citation.cfm?id=2566972.2566993>.
- Hovy E., Marcus M., Palmer M., Ramshaw L., and Weischedel R. OntoNotes: The 90% solution. *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, 2006.
- Hughes T. and Ramage D. Lexical semantic relatedness with random graph walks. *In Proceedings of EMNLP-CoNLL*, 581–589, 2007.
- Intxaurreondo A., Agirre E., De Lacalle O., and Surdeanu M. *Diamonds in the rough: Event extraction from imperfect microblog data*, 641–650. Association for Computational Linguistics (ACL), 2015. ISBN 9781941643495.
- Jiang J.J. and Conrath D.W. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, cmp-lg/9709008, 1997. URL <http://arxiv.org/abs/cmp-lg/9709008>.
- Jimenez S., Becerra C., and Gelbukh A. Soft cardinality: A parameterized similarity function for text comparison. *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, SemEval '12*, 449–453, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=2387636.2387709>.
- Jones K.S. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.



- Jurafsky D. and Martin J. *Speech and Language Processing*. Pearson, second edition, 2009.
- Kageura K. and Umino B. Methods of automatic term recognition: a review. *Terminology*, 3(2):259–289, 1996.
- Koehn P., Hoang H., Birch A., Callison-Burch C., Federico M., Bertoldi N., Cowan B., Shen W., Moran C., Zens R., Dyer C., Bojar O., Constantin A., and Herbst E. Moses: Open source toolkit for statistical machine translation. *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, 2007.
- Kruszewski G. and Baroni M. So similar and yet incompatible: Toward the automated identification of semantically compatible words. *HLT-NAACL*, 2015.
- Kusner M.J., Sun Y., Kolkin N.I., and Weinberger K.Q. From word embeddings to document distances. *ICML*, 2015.
- Landauer T.K. and Dumais S.T. A solution to plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2), 211–240, 1997.
- Lawrence Page L., Sergey Brin S., Rajeev Motwani R., and Terry Winograd T. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. URL <http://ilpubs.stanford.edu:8090/422/>. Previous number = SIDL-WP-1999-0120.
- Leacock C. and Chodorow M. Combining local context and wordnet similarity for word sense identification. In Fellbaum C., editor, *MIT Press*, 265–283, Cambridge, Massachusetts, 1998.
- Lee M.D., Pincombe B., and Welsh M. An empirical evaluation of models of text document similarity. *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, 1254–1259, Mahwah, NJ, 2005.
- Lesk M. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. *Proceedings of SIGDOC'86*, 1986.
- Li Y., McLean D., Bandar Z.A., O'Shea J.D., and Crockett K. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge*

## BIBLIOGRAPHY

---

- and Data Engineering*, 18(8):1138–1150, August 2006. ISSN 1041-4347. URL <http://dx.doi.org/10.1109/TKDE.2006.130>.
- Lin C.Y. Rouge: A package for automatic evaluation of summaries. *Proc. ACL workshop on Text Summarization Branches Out*, page 10, 2004. URL <http://research.microsoft.com/~cyl/download/papers/WAS2004.pdf>.
- Lin D. Using syntactic dependency as local context to resolve word sense ambiguity. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, 64–71, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.
- Lin D. Automatic retrieval and clustering of similar words. *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*, COLING '98, 768–774, Stroudsburg, PA, USA, 1998a. Association for Computational Linguistics. URL <http://dx.doi.org/10.3115/980432.980696>.
- Lin D. An information-theoretic definition of similarity. *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, 296–304, San Francisco, CA, USA, 1998b. ISBN 1-55860-556-8. URL <http://dl.acm.org/citation.cfm?id=645527.657297>.
- Lopez-Gazpio I., Maritxalar M., Gonzalez-Agirre A., Rigau G., Uria L., and Agirre E. Interpretable semantic textual similarity: Finding and explaining differences between sentences. *Knowl.-Based Syst.*, 119:186–199, 2017. URL <http://dx.doi.org/10.1016/j.knosys.2016.12.013>.
- Luhn H.P. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, 1(4):309–317, 1957.
- Lyon C., Malcolm J., and Dickerson B. Detecting short passages of similar text in large document collections. *PROCEEDINGS OF THE 2001 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING*, 118–125, 2001.
- Makoto N., Mikio M., and Hiroyuki I. An automatic method of the extraction of important words from japanese scientific documents. *Information pro-*

- cessing in Japan*, 16:83–88, 1976. URL <http://ci.nii.ac.jp/naid/110002672327/en/>.
- Manning C. and Schütze H. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.
- Manning C.D., Surdeanu M., Bauer J., Finkel J., Bethard S.J., and McClosky D. The Stanford CoreNLP natural language processing toolkit. *Association for Computational Linguistics (ACL) System Demonstrations*, 55–60, 2014. URL <http://www.aclweb.org/anthology/P/P14/P14-5010>.
- Marchionini G. Exploratory search: From finding to understanding. *Communications of the ACM*, 49(4):41–49, 2006.
- McCandless M., Hatcher E., and Gospodnetic O. *Lucene in Action*. Manning Publications, 2010.
- McCarthy P.M. and Jarvis S. Mtd, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42(2):381–392, 2010. ISSN 1554-3528. URL <http://dx.doi.org/10.3758/BRM.42.2.381>.
- Mihalcea R., Corley C., and Strapparava C. Corpus-based and knowledge-based measures of text semantic similarity. *Proceedings of the American Association for Artificial Intelligence (AAAI 2006)*, Boston, Massachusetts, July 2006.
- Mikolov T., Chen K., Corrado G., and Dean J. Efficient estimation of word representations in vector space. *Proceedings of Workshop at the International Conference on Learning Representations (ICLR 2013)*, abs/1301.3781 lib., Scottsdale, AZ, USA, 2013a. URL <http://arxiv.org/abs/1301.3781>.
- Mikolov T., Sutskever I., Chen K., Corrado G., and Dean J. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013b. URL <http://arxiv.org/abs/1310.4546>.
- Miller G.A., Beckwith R., Fellbaum C., Gross D., Miller K., and Teng R. Five papers on wordnet. *Special Issue of the International Journal of Lexicography*, 3(4):235–312, 1991.
- Milne D. Computing semantic relatedness using Wikipedia’s link structure. *Proceedings of the New Zealand Computer Science Research Student Conference*, 2007.

## BIBLIOGRAPHY

---

- Milne D. and Witten I. Learning to Link with Wikipedia. *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM'2008)*, Napa Valley, California, 2008.
- Mohammad S., Dorr B., and Hirst G. Computing word-pair antonymy. *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08*, 982–991, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=1613715.1613843>.
- Moldovan D. and Rus V. Logic form transformation of wordnet and its applicability to question answering. In *Proceedings of ACL 2001*, 394–401, 2001.
- Mu J., Bhat S., and Viswanath P. All-but-the-Top: Simple and Effective Postprocessing for Word Representations, February 2017. URL <http://arxiv.org/abs/1702.01417>.
- O'Donnell M., Mellish C., Oberlander J., and Knott A. ILEX: An architecture for a dynamic hypertext generation system. *Natural Language Engineering*, 7: 225–250, 2001.
- Oliva J., Serrano J.I., del Castillo M.D., and Iglesias A. Symss: A syntax-based measure for short-text semantic similarity. *Data Knowl. Eng.*, 70(4):390–405, April 2011. ISSN 0169-023X. URL <http://dx.doi.org/10.1016/j.datak.2011.01.002>.
- Ounis I., Amati G., Plachouras V., He B., Macdonald C., and Lioma C. Terrier: A High Performance and Scalable Information Retrieval Platform. *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.
- Pang B. and Lee L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008. ISSN 1554-0669. URL <http://dx.doi.org/10.1561/1500000011>.
- Pantel P. and Lin D. An unsupervised approach to prepositional phrase attachment using contextually similar words. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL '00*, 101–108, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. URL <http://dx.doi.org/10.3115/1075218.1075232>.

- Parapar D., Barreiro A., and Losada D.E. Query expansion using wordnet with a logical model of information retrieval. *Proceedings of IADIS AC*, 487–494, 2005.
- Patwardhan S., Banerjee S., and Pedersen T. Using measures of semantic relatedness for word sense disambiguation. In Gelbukh A.F., editor, *CICLing*, 2588 lib. of *Lecture Notes in Computer Science*, 241–257. Springer, 2003.
- Pedersen T., Patwardhan S., and Michelizzi J. Wordnet::similarity: Measuring the relatedness of concepts. *Demonstration Papers at HLT-NAACL 2004, HLT-NAACL–Demonstrations ’04*, 38–41, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=1614025.1614037>.
- Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., and Duchesnay E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Pennington J., Socher R., and Manning C.D. GloVe: Global Vectors for Word Representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D14-1162>.
- Pollack J.B. Recursive distributed representations. *Artif. Intell.*, 46(1-2):77–105, November 1990. ISSN 0004-3702. URL [http://dx.doi.org/10.1016/0004-3702\(90\)90005-K](http://dx.doi.org/10.1016/0004-3702(90)90005-K).
- Rapp R. Automatic identification of word translations from unrelated english and german corpora. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 11–14, Maryland, 1999.
- Rapp R. A freely available automatically generated thesaurus of related words. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, 395–398, Lisboa, Portugal, 2004.
- Rashtchian C., Young P., Hodosh M., and Hockenmaier J. Collecting image annotations using Amazon’s Mechanical Turk. *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk, CSLDAMT 2010*, 139–147, Stroudsburg, PA, USA, 2010. Association

## BIBLIOGRAPHY

---

- for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=1866696.1866717>.
- Resnick P. and Varian H. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.
- Resnik P. Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI, IJCAI'95*, 448–453, 1995. ISBN 1-55860-363-8, 978-1-558-60363-9. URL <http://dl.acm.org/citation.cfm?id=1625855.1625914>.
- Rigau G., Rodríguez H., and Agirre E. Building accurate semantic taxonomies from monolingual mrds. *Proceedings of COLING/ACL*, Montréal, Canada, 1998.
- Robert and Paris C. Using Dependency-based Features to Take the "Para-farce" out of Paraphrase. *Australasian Language Technology Workshop 2006 (ALTW 2006)*, 131–138, 2006.
- Robertson S.E. and Jones K.S. Relevance weighting of search terms. *J. Am. Soc. Inf. Sci.*, 27(3):129–146, 1976. ISSN 1097-4571. URL <http://dx.doi.org/10.1002/asi.4630270302>.
- Roes I., Stash N., Wang Y., and Aroyo L. A personalized walk through the museum: the CHIP interactive tour guide. *Proc. of the 27th International Conference on Human Factors in Computing Systems*, 3317–3322, Boston, MA, 2009.
- Rubenstein H. and Goodenough J.B. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October 1965. ISSN 0001-0782. URL <http://doi.acm.org/10.1145/365628.365657>.
- Rychalska B., Pakulska K., Chodorowska K., Walczak W., and Andruszkiewicz P. Samsung Poland NLP Team at SemEval-2016 Task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, CA, USA, 2016.
- Sahlgren M. *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional*

- Vector Spaces*. Doktoretza-tesia, Stockholm University, Stockholm, Sweden, 2006.
- Salton G. and Buckley C. Term-weighting approaches in automatic text retrieval. *Proceedings of Information Processing and Management*, 513–523, 1988.
- Salton G. and McGill M.J. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN 0070544840.
- Salton G.M., Wong A.K.C., and Yang C.S. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, November 1975.
- Sandhaus E. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 6(12), 2008.
- Schütze H. Automatic word sense discrimination. *Comput. Linguist.*, 24:97–123, March 1998. ISSN 0891-2017. URL <http://portal.acm.org/citation.cfm?id=972719.972724>.
- Schütze H. Distributional part-of-speech tagging. *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, Dublin, Ireland, 1995.
- Smadja F. Retrieving collocations from text: Xtract. *Comput. Linguist.*, 19:143–177, March 1993. ISSN 0891-2017. URL <http://portal.acm.org/citation.cfm?id=972450.972458>.
- Snover M., Dorr B., Schwartz R., Micciulla L., and Makhoul J. A study of translation edit rate with targeted human annotation. *Proceedings of Association for Machine Translation in the Americas*, 2006.
- Socher R., Huang E.H., Pennington J., Ng A.Y., and Manning C.D. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Advances in Neural Information Processing Systems 24*, 2011a.
- Socher R., Lin C.C., Ng A.Y., and Manning C.D. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. *Proceedings of ICML*, 2011b. URL <http://www.cs.stanford.edu/people/ang/papers/icml11-ParsingWithRecursiveNeuralNetworks.pdf>.



## BIBLIOGRAPHY

---

- Socher R., Manning C.D., and Ng A.Y. Learning continuous phrase representations and syntactic parsing with recursive neural networks. *In Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, 2010.
- Specia L. Exploiting objective annotations for measuring translation post-editing effort. *15th Conference of the European Association for Machine Translation, EAMT*, 73–80, Leuven, Belgium, 2011. URL <http://www.mt-archive.info/EAMT-2011-Specia.pdf>.
- Stack Exchange, Inc. Stack exchange data dump. <https://archive.org/details/stackexchange>, 2016. URL <https://archive.org/details/stackexchange>.
- Stamatatos E. Plagiarism detection using stopword n-grams. *J. Am. Soc. Inf. Sci. Technol.*, 62(12):2512–2527, December 2011. ISSN 1532-2882. URL <http://dx.doi.org/10.1002/asi.21630>.
- Stevenson M. and Greenwood M.A. Learning Information Extraction Patterns Using WordNet. *Proceedings of the 5th Intl. Conf. on Language Resources and Evaluations, LREC 2006 22 - 28 May 2006*, 2006 lib., 95–102, 2006. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.61.5378>.
- Strube M. and Ponzetto S. Wikirelate! computing semantic relatedness using wikipedia. *Proceedings of the National Conference on Artificial Intelligence*, 21 lib., page 1419. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- Sultan M.A., Bethard S., and Sumner T. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, 2:219–230, 2014a.
- Sultan M.A., Bethard S., and Sumner T. DLS@CU: Sentence similarity from word alignment. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 241–246, Dublin, Ireland, August 2014b. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S14-2010>.
- Sultan M.A., Bethard S., and Sumner T. DLS@CU: Sentence similarity from word alignment and semantic vector composition. *Proceedings of the 9th In-*



- ternational Workshop on Semantic Evaluation (SemEval 2015)*, 148–153, Denver, CO, USA, June 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S15-2027>.
- T. H. Haveliwala T.H. Topic-sensitive pagerank. *WWW '02*, 517–526, New York, NY, USA, 2002. ACM. ISBN 1-58113-449-5.
- Tai K.S., Socher R., and Manning C.D. Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, abs/1503.00075, 2015. URL <http://arxiv.org/abs/1503.00075>.
- Templin M.C. *Certain language skills in children*. University of Minnesota Press, 1957.
- Toutanova K., Klein D., Manning C., and Singer Y. Feature-rich part-of-zspeech tagging with a cyclic dependency network. *Proceedings of HLT-NAACL 2003, NAACL '03*, 252–259, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. URL <http://dx.doi.org/10.3115/1073445.1073478>.
- Turney P.D. Similarity of semantic relations. *Computational Linguistics*, 32(3): 379–416, 2006.
- Turney P.D. and Pantel P. From frequency to meaning: Vector space models of semantics. *CoRR*, abs/1003.1141, 2010. URL <http://arxiv.org/abs/1003.1141>.
- Versley Y. Decorrelation and shallow semantic patterns for distributional clustering of nouns and verbs. *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics*, 55–62, Hamburg, Germany, 2008.
- Šarić F., Glavaš G., Karan M., Šnajder J., and Dalbelo Bašić B. Takelab: Systems for measuring semantic text similarity. *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 441–448, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S12-1060>.
- Wang J., Song Y., Leung T., Rosenberg C., Wang J., Philbin J., Chen B., and Wu Y. Learning fine-grained image similarity with deep ranking. *CoRR*, abs/1404.4661, 2014. URL <http://arxiv.org/abs/1404.4661>.

## BIBLIOGRAPHY

---

- Wieting J., Bansal M., Gimpel K., and Livescu K. Towards universal paraphrastic sentence embeddings. *CoRR*, abs/1511.08198, 2015. URL <http://arxiv.org/abs/1511.08198>.
- Wilks Y., Slator B., and Guthrie L. *The Grammar of Sense: Is Word-sense Tagging Much More than Part-of-speech Tagging*. MIT Press, 1996.
- Wise M.J. Yap3: Improved detection of similarities in computer program and other texts. *Proceedings of the Twenty-seventh SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '96, 130–134, New York, NY, USA, 1996. ACM. ISBN 0-89791-757-X. URL <http://doi.acm.org/10.1145/236452.236525>.
- Xu W., Callison-Burch C., and Dolan B. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 1–11, Denver, Colorado, June 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/S15-2001>.
- Yeh E., Ramage D., Manning C.D., Agirre E., and Soroa A. Wikiwalk: random walks on wikipedia for semantic relatedness. *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-4, 41–49, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-54-1. URL <http://dl.acm.org/citation.cfm?id=1708124.1708133>.
- Zernik U. *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*. Lawrence Erlbaum Associates, 1991.
- Zipf. *Selected studies on the principle of relative frequency in language*. Harvard University Press, Cambridge, MA, 1932.