MAKING SENSE: FROM WORD DISTRIBUTION TO MEANING

ENRICO SANTUS

Ph.D

The Hong Kong Polytechnic University

2017

The Hong Kong Polytechnic University

Department of Chinese and Bilingual Studies

MAKING SENSE:

From Word Distribution to Meaning

Enrico Santus

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

May 2016

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

Enrico Santus

A Mamma, Papà, Andrea e Valeria.

E a coloro che restano.

ABSTRACT

In order to perform complex tasks, Natural Language Processing (NLP) applications need to rely on knowledge resources, whose main building blocks have been identified in *entities* and *relations* (Herger, 2014). Given their affinity to semantic memory in human beings, these resources have often been referred to as *models of semantic memory* (Jones, Willits, & Dennis, 2015).

In the last fifty years, a number of these models have been proposed in the cognitive, linguistic and computational literature (Jones, Willits, & Dennis, 2015). While the first generation models were mostly theoretical and were not designed to be computationally implemented (i.e. *classic models*), starting from the 1980s, a second generation tried to address the learnability issue by adopting representations of meaning that could be learnt automatically by observing word co-occurrence in natural text (i.e. *learning models*).

Among the second generation models, starting from the 1990s, *Distributional Semantic Models* (DSMs) gained a lot of attention in the cognitive, linguistic and computational communities because they allow the efficient treatment of word meaning and word similarity (Harris, 1954), showing furthermore consistent behaviors with psycholinguistic findings (Landauer & Dumais (1997); Lenci, (2008)). Even though these models are strong in identifying similarity (and therefore relatedness), they were found to suffer from a major limitation, that is they do not offer any principled way to discriminate semantic relations held by words. In fact,

since they define word similarity in distributional terms (i.e. *Distributional Hypothesis*; Harris (1954)), they put together, under the umbrella of similar words, terms that are related by very different semantic relations, such as *synonymy*, *antonymy*, *hypernymy* and *co-hyponymy* (Santus, Lenci, Lu, & Huang, 2015a).

In this thesis we address this limitation proposing several unsupervised methods for the discrimination of semantic relations in DSMs. These methods (i.e. *APSyn*, *APAnt* and *SLQS*) are linguistically and cognitively motivated (Murphy G. L., 2002; Cruse, 1986) and aim at identifying distributional properties that characterize the studied semantic relations (i.e. respectively, *similarity*, *opposition* and *hypernymy*), so that the DSMs are provided with useful discriminative information.

In particular, our measures analyze the properties of the most salient contexts of the target words, under the assumption that these contexts are more informative than the full distribution, which is instead assumed to include noise (Santus, Lenci, Lu, & Huang, 2015a). In order to identify the most salient contexts, for every target we sort them by either the *Positive Pointwise Mutual Information* (PPMI; Church & Hanks (1989)) or the *Positive Local Mutual Information* (PLMI; Evert (2005)), and we select the top *N* ones, which are then used for the extraction of a given distributional property (i.e. intersection, informativeness, etc.). In all our methods, *N* is a hyperparameter that can be tuned in a range between 50 and 1000.

Our measures are carefully described and evaluated, and they are shown to be competitive with the state-of-the-art, sometimes even outperforming the best models in particular settings (including the recently introduced predictive models, generally referred to as *word embeddings*; see Mikolov, Yih, & Geoffrey (2013)). Their scores, moreover, have been used as features for *ROOT9* (Santus, Lenci, Chiu, Lu, & Huang, 2016e), a supervised system that exploits a *Random Forest* algorithm to classify taxonomical relations (i.e. *hypernymy* and *co-hyponymy* versus *unrelated words*), achieving state-of-the-art performances (Weeds, Clarke, Reffin, Weir, & Keller, 2014).

The thesis is organized as follows. The *Introduction* describes the problem and the reasons behind the adoption of the distributional framework. The first two chapters describe the main models of semantic memory and discuss how computers can learn and manipulate meaning, starting from word distribution in language corpora. Three chapters are then dedicated to the main semantic relations we have dealt with (i.e. *similarity*, *opposition* and *hypernymy*) and the relative unsupervised measures for their discrimination (i.e. *APSyn*, *APAnt* and *SLQS*). The final chapter describes the supervised method *ROOT9* for the identification of taxonomical relations. In the *Conclusions*, we summarize our contribution and we suggest that future work should target i) the systematic study of the hyperparameters (e.g. the impact of *N*); ii) the merging of the methods for developing a multi-class classification algorithm; and iii) the adaptation of the methods (and/or their principles) to reduced matrices (see Turney & Pantel (2010)) and *word embeddings* (see Mikolov, Yih, & Geoffrey (2013))

PUBLICATIONS

JOURNAL

Santus, E., Lenci, A., Lu, Q., & Huang, C.-H. (2015a). When Similarity Becomes Opposition: Synonyms and Antonyms Discrimination in DSMs. In *Italian Journal on Computational Linguistics,* aAccademia University Press.

2016

Chersoni, E. Santus, E, Lenci, A., Blache, P., & Huang C.-R. (2016a). Representing Verbs with Rich Contexts: an Evaluation on Verb Similarity. Austin, Texas (USA): Proceedings of Empirical Methods on Natural Language Processing (EMNLP, 2016).

Chersoni, E., Rambelli, G., Santus, E. (2016b). CogALex-V Shared Task: ROOT18. Osaka, Japan: Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V).

Liu, H., Neergaard, K., Santus, E., & Huang, C.-R. (2016). EVALution-MAN: A Chinese Dataset for the Training and Evaluation of DSMs. Portorož, Slovenia: Proceedings of Language Resources and Evaluation Conference (LREC, 2016).

Santus, E., Chersoni, E., Lenci, A., Huang, C.-R., & Blache, P. (2016a). Testing APSyn against Vector Cosine on Similarity Estimation. Seoul, South Korea:

Proceedings of Pacific Asia Conference on Language, Information and Computation (PACLIC, 2016).

Santus, E., Chiu, T.-S., Lu, Q., Lenci, A., & Huang, C.-R. (2016b). Unsupervised Measure of Word Similarity: How to Outperform Co-occurrence and Vector Cosine in VSMs. Phoenix, Arizona: Proceedings of American Association for the Advancement of Artificial Intelligence (AAAI, 2016).

Santus, E., Chiu, T.-S., Lu, Q., Lenci, A., & Huang, C.-R. (2016c). What a Nerd! Beating Students and Vector Cosine in the ESL and TOEFL Datasets. Portorož, Slovenia: Proceedings of 10th Conference on Language Resources and Evaluation (LREC, 2016).

Santus, E., Gladkova, A., Evert, S., & Lenci, A. (2016d). The CogALex-V shared task on the corpus-based identification of semantic relations. Osaka, Japan: Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V), pages 69–79.

Santus, E., Lenci, A., Chiu, T.-S., Lu, Q., & Huang, C.-R. (2016e). Nine Features in a Random Forest to Learn Taxonomical Semantic Relations. Portorož, Slovenia: Proceedings of Language Resources and Evaluation Conference (LREC 2016).

Santus, E., Lenci, A., Chiu, T.-S., Lu, Q., & Huang, C.-R. (2016f). ROOT13: Spotting Hypernyms, Co-Hyponyms and Randoms. Phoenix, Arizona: Proceedings of Association for the Advancement of Artificial Intelligence (AAAI).

2015

Santus, E., Yung, F., Lenci, A., & Huang, C.-R. (2015b). EVALution 1.0: an Evolving Semantic Dataset for Training and Evaluation of Distributional Semantic Models. Beijing, China: Proceedings of The 4th Workshop on Linked Data in Linguistics (LDL-2015), (ACL, 2015).

Tungthamthiti, P., Santus, E., Xu, H., Huang, C.-R., & Kiyoaki, S. (2015). Sentiment Analyzer with Rich Features for Ironic and Sarcastic Tweets. Shanghai, China: Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC, 2015).

Xu, H., Santus, E., Laszlo, A., & Huang C.-R. (2015). LLT-PolyU: Identifying Sentiment Intensity in Ironic Tweets. Denver, Colorado (USA): Proceedings of the 9th Workshop on Semantic Evaluation (SemEval, 2015).

2014

Santus, E., Lenci, A., Lu, Q., & Schulte im Walde, S. (2014a). Chasing Hypernyms in Vector Spaces with Entropy. Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014), 2, p. 38-42.

Santus, E., Lu, Q., Lenci, A., & Huang, C.-R. (2014b). Taking Antonymy Mask off in Vector Space. Phuket, Thailand: Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation (PACLIC 2014).

Santus, E., Lu, Q., Lenci, A., & Huang, C.-R. (2014c). Unsupervised Antonym-Synonym Discrimination in Vector Space. Pisa, Italy: Atti della Conferenza di Linguistica Computazionale Italiana (CLIC-IT 2014).

ACKNOWLEDGEMENTS

This thesis is the result of a three-year effort, during which I could benefit from the valuable advices of my supervisors Chu-Ren Huang and Qin Lu, as well as the important help of Alessandro Lenci. My sincere gratitude goes to them for having been always present and supportive.

These pages would not have existed without the daily care of my family. I thank my parents, my siblings and my extended family for everything they taught me and for the way they constantly encourage me to achieve my goals. In particular, I wish to thank my brother and my uncle Franco, for having shown me that everything is possible with a good will.

That said, I would like to thank the friends who have been close to me during these three years, sharing happiness, sadness and anger. I can not list them all, but I would like to name a few: Davide Arosio, Valentina Campus, Emmanuele Chersoni, Jess Chu, Gil Coimbra, Leonardo Cosmai, Eddie Cheung, Giuseppe de Vito, Poe Ho, Luca Iacoponi, Maki Kawamura, Charis Kwok, Kristina Mowinska, Alessandro Pilleri, Lorenza Scano, Dominik Schlechtweg, Alessandro Senis, Vered Shwartz, Frances Yung.

Finally, a special thank goes to *The Hong Kong Ph.D Fellowship Scheme*, without which I could never have afforded to live and study in this wonderful city. I have loved learning more about it and its culture. And I have loved interacting with

its citizens, who impressed me for their welcome and their enthusiasm in protecting their democratic rights.

TABLE OF CONTENTS

Introduction

In the last decades, Natural Language Processing (NLP) has achieved impressive progress in modelling human language ability. Motivated simultaneously by engineering and scientific interests, the discipline led to the development of a large number of applications, which are able to perform almost any kind of linguistic task – including *Question Answering* (QA), *Information Extraction* (IE), *Machine Translation* (MT), *Speech Synthesis* (SS), and so on (Jurafsky & Martin, 2009; Manning & Schütze, 1999). These applications are not only useful (i.e. Google search engine and Siri's voice are used by millions of users every day), but they also provide interesting insights about how humans actually manipulate language (Jones, Willits, & Dennis, 2015).

The accuracy of these models has constantly increased not only thanks to the improvement of the algorithms, but also thanks to the adoption of richer and more powerful semantic representations (Lenci, 2010). Such representations are in fact of paramount importance for providing NLP applications with the necessary knowledge to perform complex tasks (Lenci, 2010). Since their role in NLP is similar to the one played by the semantic memory for human beings, they are generally referred to as *models of semantic memory* (Jones, Willits, & Dennis, 2015).

Given their importance, methods for their automatic development and update assumed a key role in NLP (Lenci, 2010). In particular, several studies introduced novel methods for the automatic extraction, representation and manipulation of *entities* and *relations*, which have been identified as the main building blocks of such resources (Herger, 2014).

The focus of this thesis is to suggest some principled ways to provide a specific family of models of semantic memory, namely *Distributional Semantic Models* (DSMs), with information to discriminate semantic relations. *DSMs* models have gained particular attention in the cognitive, linguistic and computational communities because of their ability of efficiently treat word meaning and word similarity (Harris, 1954), showing furthermore consistent behaviors with psycholinguistic findings (Landauer & Dumais (1997); Lenci, (2008)). However, they were found to suffer from a major limitation: since they rely on a loose definition of similarity (i.e. distributional similarity as it is defined by the *Distributional Hypothesis*; Harris (1954)), they cluster together, under the umbrella of similar words, terms that related by different semantic relations, such as *synonymy*, *antonymy*, *hypernymy* and *co-hyponymy* (Santus, Lenci, Lu, & Huang, 2015a). Our objective is therefore to propose unsupervised methods (i.e. *APSyn*, *APAnt* and *SLQS*) that aim at identifying specific distributional properties of such semantic relations (i.e. respectively, *similarity*, *opposition* and *hypernymy*) to provide DSMs with principled ways for their discrimination. These features are finally combined in a supervised model (i.e. *ROOT9*) for the multi-class classification of taxonomical relations (i.e. hypernymy and co-hyponymy).

A common assumption of our unsupervised measures is that for the identification of semantic relations the most salient contexts of the target words are more informative than their full distribution, which is instead assumed to include noise (Santus, Lenci, Lu, & Huang, 2015a). In order to identify the most salient contexts, for every target we sort them by either the *Positive Pointwise Mutual Information* (PPMI; Church & Hanks (1989)) or the *Positive Local Mutual Information* (PLMI; Evert (2005)), and we then select the top *N* ones, which will be used for the extraction of the distributional properties that seem to characterize specific semantic relations (i.e. intersection, informativeness, etc.). In all our methods, *N* is a hyperparameter that can be tuned in a range between 50 and 1000.

The thesis is organized as follows. In Chapter I, we present several *models of semantic memory*, discussing their ability of automatically learn semantic representations (i.e. word meaning). In this respect, we describe language as a *system of symbols* (Saussure, 1983) and meaning as a *system of relations* (i.e. both syntagmatic and paradigmatic ones; Murphy (2003)). We identify the distributional approach as our framework, underlining its strength in automatically representing *word meaning* and modeling *word similarity* (Harris, 1954). Nevertheless, we report its weakness in identifying specific semantic relations, leaving to the rest of the thesis the description and evaluation of methods for their identification. In Chapter II, we summarize the origin and the development of the distributional approach, discussing it from both the cognitive and linguistic perspectives (Lenci, 2008). We show, then, how to develop a *Distributional Semantic Model* from scratch, reporting and elaborating on the linguistic and mathematical steps identified by Turney & Pantel (2010). The subsequent chapters are dedicated to *similarity* (Chapter III), *opposition* (Chapter IV) and *hypernymy* (Chapter V). Each of these chapters starts with the definition of the relation, followed by the description and evaluation of our method for its identification (i.e. respectively *APSyn*, *APAnt* and *SLQS*). Chapter VI, finally, describes *ROOT9*, a supervised method for the classification of *taxonomical relations* (Santus, Lenci, Chiu, Lu, & Huang, 2016e). The thesis concludes summarizing our contribution and suggesting directions for future research, which include i) the systematic study of the hyperparameters (e.g. the impact of *N*); ii) the merging of the methods for developing a multi-class classification algorithm; and iii) the application of the methods (or their principles) to reduced matrices (see Turney & Pantel (2010)) and *word embeddings* (Mikolov, Yih, & Geoffrey (2013)).

Chapter I – Semantic Memory and Semantic Relations: From Word Distribution to Meaning

"Words are but symbols for the relations
of things to one another and to us;
nowhere do they touch upon absolute truth"
F. Nietzsche, *Philosophy in the Tragic Age of the Greeks*

In this chapter, we discuss the importance of models of semantic memory in NLP (Section 1.1), providing an overview of the principal ones, classified as *classic* (Section 1.1.1) and *learning models* (Section 1.1.2). In Section 1.2, we show how the latter address the learnability issue by observing the regularities of language and we briefly describe the connection between syntagmatic and paradigmatic relations (Section 1.2.1). After identifying the distributional approach as one of the most powerful approaches to model semantic memory, we argue that such approach suffers from the lack of a principled way to identify semantic relations. In section 1.3 and 1.3.1, therefore, we describe the importance of semantic relations, setting the description and evaluation of methods for the identification of *synonymy*, *antonymy* and *hypernymy* as major objective of the thesis. In sections 1.3.2 and 1.3.3, we also clarify the terminology that will be used in the rest of the text. We conclude the chapter with a summary in Section 1.4.

1.1 Models of Semantic Memory

Alike human beings, in order to perform any linguistic or cognitive task (retrieval, recognition, categorization, etc.), computers need to have access to resources that contain conceptual information stored in a way that can be easily identified and exploited. Such resources are often referred to as *models of semantic memory* because of their resemblance to human memory and its functions (Jones, Willits, & Dennis, 2015).

Models of semantic memory vary according to three strictly correlated factors: i) how they represent meaning (i.e. feature lists, graphs, vectors, etc.); ii) how they are implemented and updated (i.e. hand-crafted or automatically learnt in a supervised or unsupervised way); iii) how cognitively plausible and computationally efficient they are (Jones, Willits, & Dennis, 2015).

The first factor is the most important, as it affects all the others. How to represent meaning is, in fact, both a theoretical and practical problem, which has attracted the interest of philosophers, linguists, psychologists and computer scientists (Clark & Pulman, 2007). The choice of the representation has consequences on how conceptual information should be learnt, on what is stored and what is derived, and on how to account for cognitive findings and computational efficiency (Murphy M. L., 2003; Jones, Willits, & Dennis, 2015).

An ideal model of semantic memory should account for a large range of cognitive findings (i.e. it should perform like humans), while being able to efficiently learn and manipulate linguistic (i.e. phonological, morphological, syntactic and semantic; see Lenci (2010)) and non-linguistic (i.e. visual, acoustic, tactile, and so on; see Barsalou (2008)) information.

In the last fifty years, numerous models of semantic memory have been proposed (Collins & Quillian, 1969; Collins & Loftus, 1975; Rips, Shoben, & Smith, 1973; Smith, Shoben, & Rips, 1974; Osgood, 1971; McClelland & Rumelhart, 1986; Landauer & Dumais, 1997). The first generation of models (also known as *classic models*) was mostly theoretical rather than empirical, and they were not designed to automatically learn knowledge (Jones, Willits, & Dennis, 2015). When computational models inspired to this generation started being developed (e.g. WordNet has much in common with the hierarchical model proposed by Collins & Quillian (1969)), their need of being hand-crafted became an obvious limitation. For this reason, starting from the second half of the 1980s, researchers proposed a new generation of models that was finally able to learn semantic representations by exploiting natural text corpora. This new generation includes the *connectionist* and the *distributional* models (Jones, Willits, & Dennis, 2015).

In the following subsections, we briefly describe the classic (1.1.1) and the learning models (1.1.2), summarizing their strengths and limitations.

1.1.1 Classic Models

The classic models were introduced and discussed in the cognitive literature between the late 1960s and the early 1980s. They were designed as static models, representing conceptual information through one of the following representations: i) associative; ii) feature-based; iii) spatial.

The associative representation is based on the idea that concepts can be thought as nodes in a conceptual network, so that the knowledge about them is represented as the set of associations (relations) with other concepts. Such network allows simple reasoning, such as entailment (e.g. “if X is a DOG, then X is also an ANIMAL; if ANIMAL breathes, then X breathes too”).

The first model adopting an associative representation was proposed by Collins & Quillian (1969), who described it as a *hierarchical semantic network* where concepts are nodes and propositions are labeled links (see Figure 1). Such hierarchical structure represents at the same time conceptual and propositional information, complying with a principle of cognitive economy. Attributes are in fact inherited by the children nodes in the hierarchy, avoiding the need of re-stating them for every concept.

This model showed some reliability in predicting subjects’ latency in verifying statements, with a positive correlation between response time and distance of the concepts in the hierarchy. In a more careful investigation, however, Conrad (1972) showed that the strength of association between concepts was a better predictor. Also, the hierarchical model does not account for typicality (i.e. prototypical concepts are treated like un-prototypical ones) and suffers from the so-called “tennis problem” (Fellbaum C. , 1998), which is the absence of relations between terms that are not taxonomically related but that are nonetheless associated, such as *player*, *tennis court*, *racket* and *ball*. Despite the criticisms, this model inspired further research and computational models (e.g. WordNet is organized in a strict hierarchical structure).

In the attempt to address some of the limitations of the hierarchical model, Collins & Loftus (1975) deemphasized the hierarchical structure in favor of a network characterized by the idea of spreading activation. In this model (generally referred to as *spreading activation model*), any stimulus in the network is propagated to the neighboring nodes, activating them. The flexibility of this model was at the same time its strength and its weakness: while it allowed accounting for a number of behavioral phenomena, it could have potentially accounted for any data pattern (Johnson-Laird, Herrmann, & Chaffin, 1984). Nonetheless, many principles of the spreading activation model were later inherited by the connectionist models.

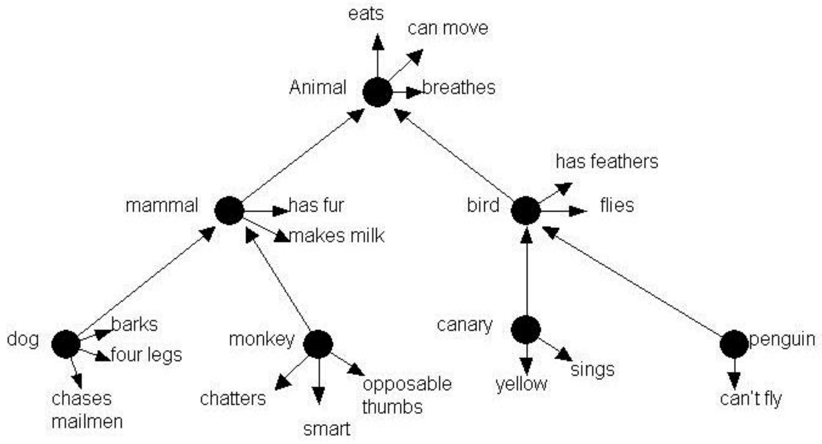

Figure 1: Hierarchical model[1]

A competing representation was proposed by Rips, Shoben, & Smith (1973). This representation is called *feature-based* because concepts are encoded as lists of

[1] Adapted from: http://images.slideplayer.com/17/5320534/slides/slide_12.jpg

binary primitives (Katz & Fodor, 1963; Murphy M. L., 2003) such as those in examples (1) and (2):

(1) *girl*: [+human, -adult, +female]

(2) *woman*: [+human, +adult, +female]

In this approach, learning a new concept consists in filling in a list of features that unambiguously identify it. Similarity and relations can be derived then from the analysis of the common and uncommon primitives. For example, we may consider (1) and (2) as holding an opposition relation (i.e. girl-woman), because they share all the features but one (i.e. the primitive *adult* has opposite sign), as suggested by the *paradox of simultaneous similarity and difference* proposed by Cruse (1986). A major problem of this approach is how to identify which semantic features are relevant for the characterization of a specific concept.

Osgood (1971) proposed a third representation method, called *spatial model of semantic memory*. This model represents concepts as vectors in a multidimensional Cartesian space, in which every dimension describes a property (e.g. weight, size, etc.) and the position of the vector in that dimension is based on human ratings about the relative property (e.g. weight: *light-heavy*, size: *small-big*, etc.). In the spatial model, similarity can be measured in terms of distance in the multidimensional Cartesian space. The spatial representation was later adapted by the distributional models.

1.1.2 Learning Models

A major shortcoming of the classic models is that they were mostly theoretical, and they were not designed to learn conceptual representations, forcing the first computational models inspired to them to be hand-crafted (Jones, Willits, & Dennis, 2015). In the second half of the 1980s, such limitation was addressed by two new types of models: the *connectionist* and the *distributional* one (McClelland & Rumelhart, 1986; Miller & Charles, 1991). Such models – which respectively inherited some characteristics from the *spreading activation* and the *spatial model* – were designed to automatically learn conceptual information (henceforth *word meaning*: see Section 1.3.2) from natural text corpora.

Connectionist models represent word meaning in terms of weighted connections between neurons in layer units (i.e. input, hidden layers and output). The weights are generally randomly initiated and then they are tuned through either supervised (by matching input and output training examples; Kohonen (1982)) or unsupervised (through the frequency of activation, as for the Hebbian Learning; Hebb (1949); Grossberg (1976)) training.

When neurons are only connected to the next layer and the activation is propagated from the input layer towards the output one, the network is referred to as *feed-forward network*. A critical issue in this representation is that the same neuron is generally used for more patterns of activation (e.g. for both *run* and *swim*) and this may cause the weight to change to better address one of the patterns, penalizing the others. These models have shown human-like behaviors (Jones, Willits, & Dennis, 2015). For example, Rogers & McClelland (2006) claimed that, during training, the internal concept representation shows progressive differentiation, learning broad distinction first and fine-grained then, similarly to what children do. An example of feed-forward network is the one trained by Rumelhart & Todd (1993) to output semantic features (e.g. *wings*) after getting in input one unit for concepts (e.g. *robin*) and one for relations (e.g. *has*). Two sets of hidden units are trained in a supervised way to match the inputs and output. Interestingly, by tuning the weights, these hidden layers are not simply learning the relations, but they are developing complex representations that can be used to achieve good performance even on unknown data.

Another typology of connectionist models, sometimes referred to as *dynamical models*, involves bi-directionality (i.e. *feedback*) and/or recurrent connectivity (Hopfield, 1982). *Dynamical connectionist models* have been used to study a numerous cognitive and neural phenomena (Jones, Willits, & Dennis, 2015). McLeod, Shallice, & Plaut (2000) proposed a dynamical model to pronounce words. It encodes orthography, phonology and semantic features of words in three different layer units, separated by additional hidden layers. Their model allows going from orthography to phonology and the other way round. Since the grapheme, sememe and phoneme layers have recurrent connections (i.e. *loops*), and since they are bi-directional, if non-words are provided (e.g. “dag”), the network searches for a stable attractor, eventually finding it in an activated neighbor (e.g. either “dad” or “dog”).

The other approach that addresses the learnability issue is the distributional one (Miller & Charles, 1991; Harris, 1954), which represents word meanings through vectors recording the frequency of co-occurrence between target words and their contexts in large corpora (Turney & Pantel, 2010). While a vector has no intrinsic meaning, except storing the information about the distribution of the target words, its relative position in the *n* dimensional semantic space can be exploited to evaluate its similarity with respect to other targets. Such similarity is generally calculated as proximity between the vectors in the vector space, often measured through the *vector cosine* (Turney & Pantel, 2010).

Within the distributional semantic framework, learning a new word meaning consists in encoding a new vector with the distributional information of the word in the corpus (Turney & Pantel, 2010). This approach was used to study and derive syntagmatic and paradigmatic information (e.g. morphological, syntactic and semantic ones). Distributional approaches were shown to perform likewise humans in predicting word similarity (Landauer & Dumais, 1997).

A second generation of distributional models, often referred to as *word embeddings*, has been recently introduced in the literature (Mikolov, Yih, & Geoffrey, 2013; Huang, Manning, & Ng, 2012; Collobert & Weston, 2008). Unlike count-based distributional models, vector representations are not learnt by counting the co-occurrence frequency, but by training a neural network to predict the contexts of a given target. These models have shown a strong ability to capture similarity and analogies, as in the famous “King - Man + Woman = Queen” example, where Mikolov and colleagues subtracted the vector of “Man” from the one of “King”, and then added the vector of “Woman”, obtaining a vector very similar to the one of “Queen”.

1.2 Learning Meaning from Word Distribution

In the previous section, we showed two families of models capable of learning word meaning by observing the word distribution in large text corpora (i.e. *connectionist* and *distributional* models). These models exploit the regularities of language to derive morphological, syntactic and semantic information (Harris, 1954). Languages are in fact complex systems of symbols and rules, and within a system "everything depends on relations" (Saussure, 1983). When humans communicate, they do not use random words in a random order, but rather they place them in the syntagmatic axis (i.e. either spoken or written text surface) in a way to fulfil certain morphological, syntactic and semantic constraints. One of the most interesting consequences of such regularities is that, assuming that there is sufficient amount of observable data, they can be used to derive paradigms, that is group of linguistic items (e.g. morphemes, lexemes, etc.) sharing similar syntagmatic behaviours at any of the linguistic levels (e.g. morphological, syntactic, semantic, etc.). In this way, computational models can explore not only how these linguistic items tend to co-occur in the linguistic surface (i.e. syntagmatic behaviour), but also how likely they are to substitute each other in the syntagm (i.e. paradigmatic behaviour), that is how similar they are to each other (Harris, 1954). By observing the syntagmatic regularities, therefore, computational models can, at least theoretically, recreate what Saussure (1983) calls the "trésor intérieur" (i.e. internal treasure), which is roughly the set the knowledge every speaker has about a language.

1.2.1 Syntagmatic and Paradigmatic Relations

Syntagmatic and paradigmatic relations are often presented in terms of axes. The syntagmatic axis (also *horizontal axis*) is generally concerned with the position in the text surface. The paradigmatic axis (also *vertical axis*) is instead concerned with the possibility of substitution.

Syntagmatic relations differ from paradigmatic ones in many aspects: i) while the former exist *in praesentia* (that is, they exist only when two words co-occur in the same spoken or written text), the latter also exist *in absentia* (that is, they exist independently of their co-occurrence in the same text, as they are part of the knowledge of speakers of a language); furthermore, ii) while syntagmatic relations involve different parts-of-speech and presuppose a certain grammatical relation (e.g. determiners generally precede adjectives or nouns), paradigmatic relations usually hold between words of the same grammatical category, since these words must be replaceable at least in some contexts.

Nevertheless, these two kinds of relations are strongly tied to each other, and such bond is at the basis of the distributional approach to word meaning, as summarized in the famous Firth (1957)'s saying "You shall know a word by the company it keeps". Miller & Charles (1991) claimed that when speakers know the meaning of a word, they do not know its dictionary definition, but rather they know how to use it, implying therefore that word meaning can be acquired by observing word usage. Evidence of distributional learning of word meanings (and of other properties) has been noted in several studies (see Ouyang, Boroditsky, & Frank

(2016)), and it was mostly attributed to the fact that words having similar meanings tend also to have a similar distributions (Harris (1954); Landauer & Dumais (1997)).

The distributional approach is adopted in this thesis as the main framework. We will show how it provides machines with a principled way to learn word meaning from word distribution and how it models word similarity (and consequently word relatedness). We will then discuss and try to address one of its major limitations: while it has been found powerful in identifying words with similar meanings (Harris, 1954), it lacks a principled way to discriminate in which way words are similar (Santus, Lenci, Lu, & Schulte im Walde, 2014c). That is, it does not offer a method for the identification of semantic relations, which are instead fundamental for the organization of the semantic memory (i.e. semantic relations contribute to many aspects of cognition, including organization and retrieval of information).

1.3 Semantic Relations

In the previous sections, we have described several models of semantic memory. We have then presented how the *learning models* – and in particular the *distributional ones* – can take advantage of large amount of linguistic data to derive paradigmatic information. We have also mentioned that the focus of this thesis will be in semantics, and more precisely in the attempt of providing DSMs with principled methods for the discrimination of semantic relations.

In this section and in its subsections we would like to provide the reader with a background on the object of research, leaving the details to the relevant chapters. We take advantage of this section and its subsections also to clarify the terminology that will be used in the following pages.

1.3.1 Lexical Semantic Relations

Semantic relations have attracted the interest of scholars from numerous fields, such as philosophy, linguistics, psycholinguistics, cognitive sciences, computer scientists, and so on. If, on the one hand, such interest has enriched the perspectives from which the phenomenon was studied, on the other hand, it has increased the terminology and its ambiguity (Murphy M. L., 2003). On top of it, every discipline has approached semantic relations with different assumptions, goals and methods. Some disciplines have based their claims on evidence in behavioural data, others on lexicographic methods and others on corpora. Unfortunately, however, rarely these different assumptions, sources and methods have led the investigators to the same conclusions. A very basic open issues is, for example, what exactly semantic relations relate: words, senses or referents in the real world? While lexical semanticists claim that they relate senses (Lyons, 1977), we cannot ignore that *dog* is the hyponym of *mammal* because the *DOG IS-A-KIND-OF MAMMAL* (where capitalization indicates entities and relationship in the real world). In this thesis, we will treat semantic relations as relating word meanings.

1.3.2 Words, Word Meanings and Concepts

Up to this moment, we have talked about relations between *words*, *word meanings* and *concepts* without clearly defining these terms. The reason is that they are difficult to pin down. The term *word*, for example, may have multiple meanings, depending on the contexts in which it is used (Matthews, 1991). Generally, *word* is used to refers to *lexical units* (e.g. “There are seven words in this sentence!”), but it can be also used to refer to *lexemes* (e.g. "*Go* and *gone* are the same *word*"). The term *word meaning*, on the other hand, is often used to refer to *concepts* and, in fact, there seems to be a large overlap between these two notions, which however does not correspond to a perfect one-to-one match (Hirst, 2009). Given their similarity, it is not rare to read in the literature experiments illustrated in terms of words and findings discussed in terms of concepts (Vigliocco & Vinson, 2007). In the same fashion, semantic relations held by words (e.g. *dog* is a hyponym of *animal*) are often used to discuss conceptual relations existing between concepts (e.g. DOG IS-A-KIND-OF ANIMAL). This is even truer in NLP, where lexical resources, such as WordNet (Fellbaum C. , 1998), are often used as conceptual ones (i.e. ontologies). However, as discussed in Murphy (2002), there might exist concepts without *words* (and therefore without *word meanings*), and we can constantly create *ad hoc* concepts for which there is no need to create a respective *word* (Barsalou, 2008). Not to mention cognitive and neurological evidence, which registers problems that only affect lexical tasks but not conceptual ones and *vice versa* (i.e. Tip of the Tongue syndrome; Murphy, 2003; Vigliocco & Vinson, 2007).

While leaving the debate about similarity and differences between *semantic* and *conceptual knowledge* to other works (see, for example, Chierchia (1997), Murphy, (2002), Murphy (2003) and Hirst (2009)), in this thesis we will consider *words* as the signifiers that are arbitrarily linked to *word meanings* (Saussure, 1983), and we will treat the learning models as able to acquire and manipulate *word meanings*, rather than *concepts*. It must be specified however that most of what we say can be eventually applied to *concepts* too.

1.3.3 Near-Synonymy, Antonymy and Hypernymy

Given that in the rest of the thesis we will mostly focus on three kinds of relations (i.e. *synonymy*, *antonymy* and *hypernymy*), in this section we briefly describe them, also clarifying some terminological choices.

By synonymy we refer to the semantic relation existing between words carrying nearly the same meaning (Cruse, 2000). Since it is unlikely that words carry exactly the same meaning (otherwise there would not be reasons for both to exist) and since such relation is gradual rather than binary, we have preferred to refer to it by *similarity*, and in some cases by *near-synonymy* (see Chapter III).

Antonymy is the relation describing lexical contrast and it can be divided into two subclasses, namely *canonical* and *non-canonical* antonymy. The former refers to the relation held by pairs such as *good-bad*, *small-big*, etc., while the latter refers to the relation held by pairs such as *green-red*, *morning-afternoon*, *small-medium*, and so on. The contrast between the latter antonyms is certainly less defined and it is more context-dependent. Since in our investigation we will include any kind of semantic contrast, in this thesis we have opted for using the term *opposition* (see Chapter IV).

Differently from synonymy and antonymy, the third relation we will deal with in this thesis – i.e. hypernymy – is not symmetric: while a hyponym is always a kind of its hypernym, the opposite is not always true (e.g. a *dog* is always a *mammal*; *a *mammal* is always a *dog*). This relation, generally also referred to as IS-A relation, is hierarchical and represents the backbone of the taxonomies (see Chapter VI) and the main organizer of the semantic memory (see Section 1.1.1).

Similarly to hypernymy, meronymy is also an asymmetric and hierarchical relation. It generally includes the *part-whole* (most prototypical), *made-of* and *membership* relationships. In this thesis we do not deal directly with such relationship, even though it appears in some experiments as relation from which we need to discriminate hypernymy (see, for example, Chapter V).

Few words should be spent, finally, for co-hyponyms and co-meronyms, which are semantic relations that respectively connect hyponyms and meronyms that descend from the same hypernym and holonym. Since these relations are “derivable”, in this thesis we will not describe any unsupervised method for their automatic identification. In any case, co-hyponyms (often also named *coordinates*) will appear in several experiments as relation from which to discriminate antonyms and hypernyms. On top of it, in Chapter VI, co-hyponyms are one of the three relations that we will try to classify with ROOT9.

1.4 Summary of Chapter I

In Section 1.1, we have seen that machines need semantic representations to perform any kind of linguistic task. Such representations are generally stored in resources that can be called models of semantic memory for their affinity with human memory and its function. Several models, with their specific representations, have been described in Section 1.1.1 and 1.1.2, paying particular attention to the ability to automatically learn word meaning.

Given the ability of machines in elaborating symbols and given that language is a complex system of symbols and rules, we have showed that it is possible to derive word meaning by observing word distribution in large text corpora (Section 1.2). That is, it is possible to extract paradigmatic information from syntagmatic one (Section 1.2.1). We said that the distributional framework allows to automatically learn word meaning and model word similarity (i.e. words with similar distribution are assumed to have similar meaning). This framework however suffers from a major limitation, which is the inability of identifying the semantic relations connecting similar words. We have therefore mentioned that addressing such limitation will be the main objective of this thesis and we have provided a general background about semantic relations and related terminology (Section 1.3).

In the next chapter, we describe the origins of the distributional approach, discussing its theoretical background. The chapter will also provide an overview on how to implement a Distributional Semantic Model from scratch.

Chapter II – Building a Count-Based Distributional Semantic Model

“You take the blue pill - *the story ends, you wake up*
in your bed and believe whatever you want to believe.
You take the red pill - *you stay in Wonderland*
and I show you how deep the rabbit-hole goes.”
Morpheus, *Matrix*

In this chapter, we describe the origins and development of the distributional approach (Section 2.1), discussing its linguistic and cognitive background as well as the two major interpretations of the *Distributional Hypothesis* (Harris (1954); Section 2.1.1). In Section 2.2, we discuss how the distributional approach has been adopted in NLP, listing the main parameters that need to be considered when implementing a distributional model (Section 2.2.1). The second part of the chapter deals with how to implement a *Distributional Semantic Model* from scratch (Section 2.3), describing all steps, from the data collection (Section 2.3.1) to the linguistic (Section 2.3.2) and the mathematical (Section 2.3.3) processing, as they were identified in Turney and Pantel (2010).

2.1 Origins and Development of the Distributional Approach

Although the distributional approach to semantics has mainly flourished in the last three decades, especially under the pressure of corpus linguistics, its roots can be found already in the beginning of the XX century, when Ferdinand de Saussure (1983) described language as symbolic system, stressing the importance of the bond between syntagmatic and paradigmatic relations (see Section 1.2.1).

The formalization of the distributional approach to semantics begun with the American structuralists, and in particular with Zellig Harris, who analysed linguistic expressions with a particular attention to their contexts (Harris, 1954). In his analysis of the semantic level, Harris noted that words with similar syntagmatic relations were also semantically similar. This idea was formalized in the *Distributional Hypothesis* (Harris, 1954; Miller & Charles, 1991), according to which: i) at least some aspects of the meaning of a lexical expression depend on its distribution; ii) the degree of similarity between two linguistic expressions is a function of the similarity of their contexts (Lenci, 2008).

In the first two decades, it progressed very little as it had to deal with the generative paradigm, which emphasized the importance of the competence over the performance (Lenci, 2008). It fared no better under the cognitive paradigm and the formal models of language. The former stressed a conceptualist view of semantics, often grounded on embodied representations (Barsalou, 2008), while the latter adopted a denotational approach (Lenci, 2008). In both cases, meaning was conceived as depending on external entities, and it could have not been conceived in terms of language-internal word distributions (Lenci, 2008).

Despite these difficulties, the distributional approach was still favoured by the corpus linguistics community, which was interested in the concrete evidence of linguistic data. When machines reached a sufficient computational power, the computational community started considering this quantitative approach as a powerful and efficient way to represent word meaning.

At the beginning of the 1990s, the distributional approach also found support in several psychological and cognitive studies (Lenci, 2008). Miller & Charles (1991) hypothesized that we learn semantic representations from contextual information, where by contextual information they referred not only to the linguistic co-text (e.g. linguistic items surrounding the target word), but also to the extra-linguistic context (e.g. speaker, hearer, situation, etc.). In the same period, other researchers proved the importance of distributional information for learning non-experienceable terms, such as colours and visual perception verbs in congenitally blind individuals (Landau & Gleitman, 1985; Lenci, Baroni, Cazzolli, & Marotta, 2013). Finally, neo-behaviourist psychologists noted that distributional information was suitable to model psychological phenomena, such as similarity judgements, semantic and associative priming, and so on (for an overview, see Schulte im Walde & Melinger (2008)).

2.1.1 Weak and Strong View

There are two main ways to conceive the *Distributional Hypothesis* (Harris, 1954): the weak and the strong view.

The weak view is based on the idea that word meaning determines word distribution. Thus, according to this view, word distribution can be used to derive some information about word meaning (Lenci, 2008), since between the two there is a certain correlation. This view is compatible with most of the theoretical research in linguistics.

The strong view, instead, consists in a comprehensive cognitive hypothesis about the form and origin of semantic representations (Lenci, 2008) and has its most relevant supporters in Miller & Charles (1991) who suggested that word distribution has a causal role in the formation of the semantic representations. In their theory, however, the authors consider not only the linguistic context (or co-text), but also the extra linguistic one (or context; e.g. participants, communicative situation, visual features, etc.).

Following the majority of the current computational approaches to word meaning, in the rest of the thesis we exclusively consider linguistic information, being aware that this is an approximation of the enormous amount of contextual information that might be considered. The reasons behind this choice are mostly pragmatic: computational techniques to identify, extract and manipulate linguistic information are in fact much more advanced than those for the manipulation of extra-linguistic information (i.e. audios, images, videos, etc.). This methodology does not preclude any future integration of other types of contextual information, which might enrich semantic representations.

2.2 Distributional Approach in NLP: Vectors and DSMs

The distributional approach to word meaning has been implemented in NLP by means of Vector Space Models (VSMs), which – given their scope – are generally referred to as *Distributional Semantic Models* (DSMs). These models represent linguistic expressions (in our case, words) through vectors in vector spaces (also called *semantic spaces*), similarly to the space *model of semantic memory* described in 1.1.1.

A vector is a mathematical structure which can be used to represent a target by mean of the values stored in its *n* dimensions, each of which describes a property of the target. In the case of DSMs, the dimensions are generally initialized to represent the distribution of the target among its *n* contexts. The values stored in these dimensions establish the position of the vector in an *n*-dimensional Cartesian space, and such a position is fundamental to measuring the similarity between the targets. In fact, targets that have similar meaning are likely to be characterized by a similar distribution, and in turn by a similar vector. Since similar vectors are expected to be close in the *n*-dimensional Cartesian space, the distance between vectors can be used to measure their similarity (Turney & Pantel, 2010). Interestingly, being math structures, it is possible to apply several algebraic operations on vectors, such as sum, subtraction, multiplication, division, and so on.

DSMs were firstly used for Information Retrieval (IR), with the SMART Information Retrieval System created by Salton, Yang, & Yu (1975). This system was used to retrieve documents (i.e. *targets*), which were represented as bags of words (Turney & Pantel, 2010), that is as vectors encoding the frequency of words (i.e. the properties that characterize the documents). Thanks to this representation, similar documents were represented by similar vectors, which were thus close to each other in the vector space. Identifying a specific document was therefore done by turning a query into the vector of a pseudo-document, and finding its neighbour(s) (Turney & Pantel, 2010).

The success of the SMART System encouraged further research in DSMs. It was noticed that by simply modifying the definitions of target and properties it was possible to represent not only documents, but also words, word-pairs and any other linguistic expressions.

2.2.1 Parameters: Targets, Contexts, Matrix and SoA

DSMs can be used to represent any kind of linguistic expression, by simply varying some parameters. Every DSM is defined by the quadruple <T, C, M, S> (Lowe, 2001), where i) T stands for the set of targets that constitutes the vector space; ii) C stands for the set of contexts that defines the dimensions of the vector space (and, therefore, the dimensions of the vectors); iii) M stands for the matrix in which data is stored (e.g. whether or not it is reduced); and iv) S stands for the strength of association (SoA) between the chosen targets and the contexts (i.e. frequency, *Positive Pointwise Mutual Information*, *Positive Local Mutual Information*, etc.).

Every parameter of the quadruple can be set in a different way. The set of targets (T), for example, can include words, pairs, triplets or any other item we may want to represent. The set of contexts (C), on the other hand, can be defined in terms of documents, windows of *n* content words (Lund & Burgess, 1996), dependency based contexts, joint-based contexts (Chersoni, Santus, Lenci, Blache, & Huang, 2016a), or any other contextual environment in which the targets may occur. The more complex the definition of targets and contexts, the sparser the matrix will be. The other two parameters define the shape of the matrix (M) and its actual content (S), so as to determine which algebraic and statistical operations can be applied to it. The matrix (M), in particular, describes what we have in the row and what in the columns, which can be the result of a dimensionality reduction process (e.g. *Singular Value Decomposition*, SVD; see Turney & Pantel (2010)). The parameter S, instead, defines the statistical measure that is used for quantifying the association between the rows and the columns, that is between the set of targets and the set of contexts (e.g. frequency, *Positive Pointwise Mutual Information*, *Positive Local Mutual Information*, etc.; Church & Hanks (1989); Evert (2005)).

2.3 Implementing a DSM

DSMs are implemented through a series of complex linguistic and mathematical processes, starting from large text corpora containing millions of words. In this section, we go through the major steps, as they were identified in Turney & Pantel (2010), providing additional information where relevant.

2.3.1 Corpus

Among the most popular corpora in Natural Language Processing there are the British National Corpus (BNC), the Reuters Corpus Volume 1 (RCV1), Wikipedia corpus and ukWac (Baroni, Bernardini, Ferraresi, & Zanchetta, 2009). These corpora vary in many aspects, including their content (i.e. news, articles, written and/or spoken language) and their size.

The BNC, for example, is the smallest of the abovementioned corpora and it is designed to represent the twentieth century British English. It contains 100 million words from a wide range of written and spoken sources (Aston & Burnard, 1998). It can be freely downloaded from http://www.natcorp.ox.ac.uk/. We have not used this corpus in our research, but it is often concatenated to other corpora in related work.

RCV1, instead, was released in 2000 as a collection of Reuters News stories stored in 806,791 XML files, of approximately 3.7Gb (Lewis, 2004). It contains about 150 million words. The corpus is available on request from the National Institute of Science and Technology (NIST). More information can be found at http://about.reuters.com/researchandstandards/corpus/index.asp.

Wikipedia is one of the most popular corpora in distributional semantics for several reasons. First of all, it is considered well balanced, since as an encyclopedia it treats many different domains using the most appropriate words. Second, its dimension is several times bigger than the RCV1, including more than 1 billion words, and is constantly growing. Wikipedia can be freely downloaded from https://en.wikipedia.org/wiki/Wikipedia:Database_download. Preprocessed versions of the Wikipedia corpus for several languages (English, Italian, French and German) are available at http://wacky.sslmit.unibo.it/doku.php (Baroni, Bernardini, Ferraresi, & Zanchetta, 2009). Once the corpus is obtained, if it is not preprocessed, the first step consists in cleaning out the XML markups. There are several tools to perform this task. For some of our experiments we have used the WikiExtractor tool, provided at https://github.com/jodaiber/Annotated-WikiExtractor. Such tool is a simple python script that goes through the corpus and cleans out all the markups, leaving the links. Links were then removed by using the following regular expression to identify them: '<[^>]+>'.

Another relevant corpus is ukWac, which can be also downloaded at http://wacky.sslmit.unibo.it/doku.php, already lemmatized and POS-tagged with TreeTagger. ukWaC contains 2 billion words and it is crawled from the Web, limiting the crawl to the .uk domain and using medium-frequency words from the BNC as seeds. This corpus is unfortunately very noisy and it is mostly used concatenated to the more balanced Wikipedia. In most of our experiments we have in fact adopted such concatenation.

2.3.2 Linguistic Processing

Once the raw and clean data is available, before creating the matrix, it is necessary to perform some linguistic processing (Turney & Pantel, 2010), which is generally language-dependent[5]. Linguistic pre-processing, often simply referred to as pre-processing, includes the following steps: i) tokenization; ii) normalization; and iii) annotation.

[5] While several of our DSMs were developed on the already annotated corpora, for some experiments, we have annotated the corpora ourselves with POS tags and dependency tags by using *Spacy*, a library for python that can be found at https://spacy.io/. Spacy also performs case folding and lemmatization.

2.3.2.1 Tokenization and Stop Words

The corpus consists of sentences containing words separated by spaces and/or punctuation marks. The tokenization process is needed to identify and isolate the words from each other and from other intervening characters (i.e. punctuation marks).

An accurate tokenizer must handle exceptions, such as those involving apostrophes (e.g. *don't* versus *do not*), hyphens (e.g. *above-mentioned* versus *above mentioned*), and so on (Manning & Schütze, 2008). Moreover, it should be able to identify multi-word expressions, such as *Bill Clinton* and *personal computer* (Turney & Pantel, 2010).

A step that is related to the tokenization process is the removal of stop words, which are high frequency words containing very little semantic information (e.g. determiners, prepositions, etc.). A list of 571 stop words can be found in the source code for the SMART system (Salton, Yang, & Yu, 1975).

2.3.2.2 Normalization

A second linguistic step is the normalization of the strings that differ on the surface but convey the same meaning (Turney & Pantel, 2010). For example, *is*, *are*, *was* and *were* have the same meaning, exactly as *Dog* and *dog* or *Leonardo* and *Leonardo da Vinci* have.

In order to reduce the superficial variations, several techniques can be adopted, including case folding (i.e. remove the case differences), lemmatization (i.e. reduce the inflected words to the lemma), named entity recognition and anaphora resolution (i.e. reduce different instantiations of the same entities). All these processes imply a certain amount of risk. For example, when removing case differences, attention should be paid to proper names that are also common nouns (e.g. *president Bush* versus a *bush*).

The normalization process increases recall (i.e. it reduces data sparseness) and reduces precision (Kraaij & Pohlmann, 1996). The extent of the normalization process should therefore depend i) on the goals of the system that is being built, and ii) on the size of the corpus. A small corpus, in fact, would need an extensive normalization in order to allow a sufficient recall, while a large corpus might not need it at all (Hull, 1996).

2.3.2.3 Annotation: POS Tagging, Parsing and Other Information

While different strings can convey the same meaning, it is also possible that similar strings convey different meanings (Turney & Pantel, 2010). This is, for example, the case of ambiguous words (e.g. *run* used as a noun or as a verb, or *bank* used with its different meanings). Given that words out of context lose disambiguating information, such information may need to be annotated to keep track of their original meaning.

Annotations can be done at different linguistic levels. For example, i) Part-Of-Speech tagging (POS) provides morphosyntactic information; ii) word sense tagging provides semantic information; and iii) parsing provides syntactic information.

Being the opposite of normalization, annotation keeps track of the differences, and therefore it increases precision and lowers recall (therefore increasing data sparseness; Hull (1996)). Also in this case, the amount of annotation should be evaluated according to the purpose of the system and the size of the used corpora.

2.3.3 Mathematical Processing

When the corpus has been cleaned, normalized and annotated, a series of mathematical processing is needed in order to build the DSM. This elaboration consists in: i) generating the frequency matrix; ii) adjusting the weights, so that they can better represent word distribution; iii) smoothing the matrix, in order to reduce the noise and the sparseness (Lowe, 2001; Turney & Pantel, 2010), consequently improving the calculation efficiency.

2.3.3.1 Frequency Matrix and Weights

A frequency matrix stores the frequencies of certain events. In DSMs, it stores the frequency of co-occurrence between the targets and the contexts. In order to build a frequency matrix, the system must count how many times the events take place, that is how many times a specific target and a specific context co-occur in a corpus. A way to do it is to sequentially scan the corpus, recording the events and their frequencies in a hash table that can then be used then to build the matrix (Turney & Pantel, 2010).

Frequency, however, is not very informative. There are words that are very frequent, but contain very little meaning (i.e. the stop words), as well as words that are very informative (i.e. content words) but have low frequency. In order to avoid such frequency bias and increase DSMs performance, several weighting systems can be adopted (Church & Hanks, 1989; Evert, 2005; Turney & Pantel, 2010).

Since in *Information Theory* an unexpected event is generally more informative than an expected one (Shannon, 1948), weights can be used to moderate the frequency bias towards very frequent events, producing a more informative matrix. One of the most common weights is the *Pointwise Mutual Information* (PMI), or the *Positive PMI* (PPMI; Church & Hanks (1989)), in which negative values are replaced with zeros.

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} f_{ij}} \qquad 1$$

$$p_{i*} = \frac{\sum_{j=1}^{n_c} f_{ij}}{\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} f_{ij}} \qquad 2$$

$$p_{*j} = \frac{\sum_{i=1}^{n_r} f_{ij}}{\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} f_{ij}} \qquad 3$$

$$pmi_{ij} = log_2\left(\frac{p_{ij}}{p_{i*}p_{*j}}\right) \qquad 4$$

$$ppmi_{ij} = \begin{cases} ppmi_{ij} & if\ ppmi_{ij} > 0 \\ 0 & otherwise \end{cases} \qquad 5$$

where f_{ij} is the frequency of the co-occurrence of the target i and the context j; p_{ij} is the probability that target i occurs in context j (see Equation 1); p_{i*} is the probability of the target i (see Equation 2); and p_{*j} is the probability of the context j (see Equation 3). $p_{i*}\ p_{*j}$ represents the probability of random co-occurrence. Therefore, if target and context are statistically independent, when $p_{i*}\ p_{*j} \geq p_{ij}$ we expect $PMI \leq 0$, because $log_2(0 \leq x \leq 1) \leq 0$. If instead there is a relation, we can expect that $p_{ij} > p_{i*}\ p_{*j}$, and therefore the PMI should be positive (see Equation 4). The PPMI is self-explanatory (see Equation 5).

PMI and PPMI are biased towards infrequent events. Several corrections have been proposed. Among them, the *Local Mutual Information* (LMI; Evert (2005)) and the *Positive Local Mutual Information* (PLMI), which are simply the co-occurrence frequency multiplied respectively by PMI or PPMI, in order to normalize it.

$$lmi_{ij} = f_{ij} \times pmi_{ij} \qquad 6$$

$$plmi_{ij} = f_{ij} \times ppmi_{ij} \qquad 7$$

2.3.3.2 Smoothing the Matrix: SVD and Context Selection

Smoothing the matrix consists in creating an approximating function that attempts to capture the important information, while leaving the noise out. This is generally done to increase the computational performance.

There are many different ways to smooth a matrix. One of the most popular and sophisticated methods is the *Singular Value Decomposition* (SVD; Landauer & Dumais (1997)), which is based on linear algebra and consists in decomposing a matrix M_{mn} in the product of three matrices $U_{mm} * \Sigma_{mn} * V_{nn}^T$, where U and V are in column orthonormal form (i.e. the columns are orthogonal and have unit length, $U^TU = V^TV = I$) and Σ is a diagonal matrix of singular values in decreasing order (Turney & Pantel, 2010). If only the top-*k* values are kept, it is possible to obtain $M_k = U_k * \Sigma_k * V_k^T$ which is a matrix of rank *k* that best approximates the original matrix M_{mn}. SVD has been regarded as a method for noise reduction and for the discovery of latent dimensions of meaning (i.e. Landauer & Dumais (1997) have shown that SVD allows the discovery of high-order co-occurrence, that is words appearing in contexts which are similar to each other), and it has been shown to improve similarity measurements.

Other methods include the selection of features according to minimum thresholds (e.g. minimum frequency or minimum number of targets to co-occur with) or according to a maximum acceptable quantity of features (i.e. only the top *N* features). The methods described in this thesis, rely on the most salient contexts of the target words, rather than on their full distribution (with saliency defined as the ranking in a PPMI/PLMI sorted context list). Such approach comes from the assumption that not all contexts are equal, and that the most salient contexts tend to be more informative, therefore better representing the semantics of words (see Section 3.1.1 for more information).

From a cognitive perspective, this is consistent with the findings of Smith and colleagues (Smith, Shoben, & Rips, 1974), who have noticed that some features were listed more frequently than others when subjects were asked to list features describing certain concepts, showing *de facto* that some features are more relevant than others. In the same fashion, from the distributional point of view, it is very likely that some contexts are more relevant, and therefore more informative about a specific word. On top of it, relying on a limited number of contexts also complies with the *principle of cognitive economy* (Collins & Quillian, 1969).

In our methods, we sort the contexts of every target word by either PPMI, PLMI or LMI. Since their ranking somehow resembles their relevance for the target word, we then select the top *N* contexts, where *N* is a hyperparameter empirically set in a range between 50 and 1000 (Santus, Lenci, Lu, & Schulte im Walde, 2014a). Once the most relevant contexts of the target words have been identified, they can be used to extract properties aimed at discriminating semantic relations (i.e. *hypernymy*, *co-hyponymy*, *synonymy* and *antonymy*).

2.4 Summary of Chapter II

In this chapter we have described the origin and development of the distributional approach (Section 2.1), illustrating the importance of such approach in NLP (Section 2.2) and how it can be implemented computationally (2.3). A large part of the chapter has shown the main parameters that need to be considered when developing a DSM (Section 2.2.1), as well as the linguistic (Section 2.3.2) and mathematical (Section 2.3.3) pre-processing steps that need to be performed, as they were identified in Turney and Pantel (2010).

In the next chapter, we will discuss the concept of similarity and we describe our contribution to its automatic identification.

Chapter III – Similarity

"No entity without identity"

W. v. O. Quine, *Ontological Relativity and Other Essays*

This chapter describes the concept of semantic similarity (Section 3.1) and its treatment in NLP (Section 3.2), with a particular focus on distributional semantics. While the first part of the chapter is mostly theoretical, the second part describes several distributional measures and introduces *APSyn*, an unsupervised method for similarity identification (Section 3.3). *APSyn* is evaluated on all the most popular test sets – TOEFL, ESL, MEN, WordSim-353 and SimLex-999 –, showing comparable or even better performances than state-of-the-art methods, including *word embeddings* models. Section 3.3 reports and discusses the results of the evaluation. Section 3.3.6, in particular, shows that *vector cosine* on the top-*N* contexts cannot reach the results obtained by *APSyn*, suggesting that the measure captures a different aspect of similarity.

The chapter is an adaptation and extension of:

- Santus, E., Chersoni, E., Lenci, A., Huang, C.-R., & Blache, P. (2016a). Testing APSyn against Vector Cosine on Similarity Estimation. Seoul, South Korea: Proceedings of Pacific Asia Conference on Language, Information and Computation (PACLIC, 2016).

- Santus, E., Chiu, T.-S., Lu, Q., Lenci, A., & Huang, C.-R. (2016b). Unsupervised Measure of Word Similarity: How to Outperform Co-occurrence and Vector Cosine in VSMs. Phoenix, Arizona: Proceedings of American Association for the Advancement of Artificial Intelligence (AAAI, 2016).
- Santus, E., Chiu, T.-S., Lu, Q., Lenci, A., & Huang, C.-R. (2016c). What a Nerd! Beating Students and Vector Cosine in the ESL and TOEFL Datasets. Portorož, Slovenia: Proceedings of 10th Conference on Language Resources and Evaluation (LREC, 2016).

3.1 Similarity

The interest to the concept of similarity dates back to Plato’s problem, which questions how people can acquire so much knowledge from so little information (Landauer & Dumais, 1997). During the centuries, the relevance of such relation has motivated a lot of research and discussion in many disciplines, such as linguistics, philosophy, cognitive science, and so on. Nowadays, its definition is still “liquid” (Murphy M. L., 2003), as “there is no unique answer to the question of how similar is one object to another” (Murphy & Medin, 1985).

Although the difficulty of defining similarity has led some scholar to argue that theories based on it are based on nothing (Murphy & Medin, 1985), similarity is known to play a fundamental role in categorization (Murphy G. L., 2002). The only way something can be properly categorized, in fact, is by comparing it to previously encountered examples (Murphy G. L., 2002). Such comparison can be modeled in many ways, depending on the studied object and on the adopted representation. In all cases, it has been understood as sharing some features (i.e. properties, relations, etc.; Rips, Shoben, & Smith, (1973)). In distributional semantics, where the objects of study are linguistic expressions (e.g. words) and the features are contexts, similarity has been often conceived in terms of co-occurrence. Following the *Distributional Hypothesis* (Harris, 1954), two linguistic expressions that occur in similar contexts have similar meanings[6].

This definition of similarity has three main implications. First of all, similarity has not to be seen as a binary relation, but rather as a continuum. Such continuum has at the extremes: i) words sharing all contexts (i.e. perfect synonyms, or identities) and ii) words sharing no contexts (i.e. perfectly unrelated words). Since it is not possible that two different words share all contexts (in that case, there would not be reasons for both to exist), perfect synonyms hardly exist. This is why most of the literature generally talks about near-synonyms rather than synonyms (Cruse, 1986; Murphy M. L., 2003). A second implication is that similarity is not always transitive. In fact, if we think about similarity in terms of intersections between sets of features (each set characterizing one word), it is possible that word meanings can be similar in different ways, as they may share different subsets of features. For example, if A and B are similar, and B and C are similar, we cannot conclude that A and C are also similar: consider A to be *calculator*, B to be *tool* and C to be *screwdriver*; A is

[6] It may be important to notice that some literature has discriminated between two types of similarity (Nakov & Kozareva, 2011): i) *attributional similarity*, if there is correspondence between attributes; ii) *relational similarity*, if there is correspondence between relations.

similar to B, and B is similar to C, but A and C are definitely less similar than the previous ones. A third implication is that, being similarity an intersection of features, it is symmetric: A and B share exactly the same features of B and A. This property does not hold for relations like hypernymy and meronymy, which are asymmetric, even though they are still characterized by a considerable amount of similarity (Cruse, 2000).

3.1.1 Features Salience

In compositional semantics and cognitive research, it is well known that features describing word meanings do not have equal salience. In early experiments carried out by Smith and colleagues (Smith, Shoben, & Rips, 1974), it was noticed that some features were listed more frequently by the subjects. The authors claimed that heavier weights should be assigned to those features, as they were more prototypical and therefore fundamental for the definition of a concept, while the others simply characterize it.

Interestingly, feature salience can change depending on the context (Murphy M. L., 2003). Tversky (Features of similarity, 1977), for example, noted that subjects considered *Austria* similar to *Sweden* when the competitors were *Hungary* and *Poland*, but they considered it more similar to *Hungary* when the other possible competitors were *Sweden and Norway*. He claimed that second set of words caused the geographic attribute to be more salient and discriminative.

3.2 Word Similarity in NLP

Word similarity is one of the most important and most studied problems in NLP, as it is fundamental for a wide range of tasks, such as *Word Sense Disambiguation* (WSD), *Information Extraction* (IE), *Paraphrase Generation* (PG), as well as the automatic creation of semantic resources.

As we have seen in the previous chapter, thanks to their representation, DSMs are suitable for the calculation of similarity, which is generally measured in terms of vector proximity. The vector cosine is usually adopted for such measurement, as it computes the angular width between the vectors. A number of other measures have been introduced during the last decades to address some of the limitations of vector cosine. We briefly describe some of the most known ones in 3.2.1. The chapter is then dedicated to introduce our new metric *APSyn*, which is inspired at *Average Precision* (AP; see Kotlerman, Dagan, Szpektor, & Zhitomirsky-Geffet (2010)), that computes the weighted intersection between the top most salient contexts of the words in a pair.

While we will work on “count-based” DSMs (see Chapter II), word embeddings have also been recently adopted for the study of word similarity. These models are based on the context-prediction and learn word representation through neural network training (Collobert & Weston, 2008; Mikolov, Yih, & Geoffrey, 2013; Huang, Manning, & Ng, 2012): word vector dimensions are set to maximize the probability for the contexts which typically occur with the target word.

Collobert & Weston (2008) proposed a convolutional neural network that, given a sentence, returns a large amount of linguistic predictors, such as POS-tags, named entity tags, semantic roles, semantically similar words, etc. Almost all tasks rely on labeled data. Huang, Manning, & Ng (2012) proposed a neural network that learns word embeddings to maximize the likelihood of predicting the last word in a sentence, relying on the local context (i.e. previous word) and on the global one (i.e. document in which the word occurs). Mikolov, Yih, & Geoffrey (2013) proposed two efficient embedding algorithms, the Skip-Gram and the Continuous-Bag-of-Words, which also learn word representations by means of neural network training, but without using a non-linear hidden layer, thus minimizing the computational complexity of the training. One of these models, the Skip-Gram with negative-sampling training method, achieves state-of-the-art results in a wide range of NLP tasks.

Extensive evaluations comparing the two families of models have given contrasting results. The first systematic comparison was carried out in a study by Baroni, Dinu, & Kruszewski (2014), in which word embedding models emerged as the clear winners. By contrast, a following paper by Levy, Goldberg, & Dagan (2015) suggested that the superiority of context-predicting models was tied to the optimal choice of some hyperparameters that had been already tuned by the algorithm designers and were not taken into account in the work by Baroni and colleagues. Their final claim was that such settings have a significant impact on the performance and that part of these parameters can be transferred to count-based models to obtain comparable results.

Other approaches to word similarity have relied on bilingual corpora, paraphrases and knowledge resources, such as lexicons or semantic networks (i.e.

WordNet; Fellbaum (1998)). These approaches achieve high precision but suffer from low recall, because of the limited coverage of the underlying resources, caused by their development and update costs (Santus, Lenci, Lu, & Huang, 2015a).

3.2.1 Similarity Measures

As we have seen in Chapter II, once the words are represented as vectors in a vector space, similarity can be calculated as proximity (Turney & Pantel, 2010). Several measures have been adopted for this purpose, including the Manhattan Distance (L1), which is the sum of the differences between the vectors' dimensions, and the Euclidean Distance (L2), which is the squared root of the sum of the squared differences between the vectors' dimensions (Deza & Deza, 2009). All of them can be used and converted in measure of similarity by applying either the inversion or the subtraction (Manning & Schütze, 1999), as shown below:

$$sim(x, y) = \frac{1}{dist(x,y)} \qquad 8$$

$$sim(x, y) = 1 - dist(x, y) \qquad 9$$

A wide range of other similarity measures – such as the Dice's Coefficient, the Jaccard Similarity and the Matching Coefficient – has been adopted in the literature. However, the most common measure of word similarity in DSMs is the vector cosine (Turney & Pantel, 2010), which looks at the normalized correlation between the dimensions of two word vectors, w_1 and w_2. It is described by the following equation:

$$cos(w_1, w_2) = \frac{\sum_{i=1}^{n} f_{1i} \times f_{2i}}{\sqrt{\sum (f_{1i})^2} \times \sqrt{\sum (f_{2i})^2}} \qquad 10$$

where f_{x_i} is the *i*-th dimension in the vector *x*. Cosine ignores the length of the vectors, focusing on the angle between them (i.e. not much importance is given to the absolute frequency of the word). It returns values ranging between -1, when the vectors point to opposite directions (i.e. θ = 180°), and +1, when the vectors point to the same direction (i.e. θ = 0°), having value 0 (zero) when the vectors are orthogonal (i.e. θ = 90°). If no smoothing has been applied, frequencies cannot be negative, and therefore *vector cosine* scores can only range between 0 and 1. This measure has been extensively used to quantify word similarity in vector spaces becoming a sort of *de facto* standard in distributional semantics (Landauer & Dumais, 1997; Mikolov, Yih, & Geoffrey, 2013; Levy, Goldberg, & Dagan, 2015).

In this chapter, we introduce *APSyn*, a rank-based measure of word similarity that was shown to outperform the vector cosine in the TOEFL[7] and ESL[8] test sets

[7] The TOEFL dataset consists of 80 multiple-choice synonym questions, in which, given a target word, the system has to choose the synonym among four possible choices. After its first use in Landauer & Dumais (1997), who achieved a score of 64.38% (which is very close to the reported average of non-English US college applicant: i.e. 64.50%), the TOEFL dataset became one of the most common benchmarks for DSMs testing. Bullinaria & Levy (2012) even achieved 100% accuracy on this dataset. In their paper, the authors extensively analyze numerous parameters, including the influence of corpus size, window size, stop-lists, stemming and SVD, until they find a perfectly optimized model. After achieving perfect precision on the TOEFL, the authors acknowledge that while these results are impressive for the benchmark, they can hardly be generalized to new tasks.

[8] The ESL was proposed by Turney (2001) as a benchmark for the evaluation of systems in the identification of synonyms. It consists of 50 multiple-choice synonym questions, which are provided in a context to facilitate sense disambiguation. The best reported corpus-based approaches in this benchmark were those of Turney (2001) and Terra & Clarke (2003), while the best performing algorithm was developed by Jarmasz & Szpakowicz (2003) and relies on a thesaurus, achieving 82% of accuracy.

(Santus, Chiu, Lu, Lenci, & Huang, 2016a-b-c). This measure is based on the hypothesis that similar words not only share many contexts, but they share their most relevant ones in higher proportion to less similar words. Such a hypothesis relies on the observation that features have different salience in characterizing word meaning, as described in 2.3.3.2 and in 3.1.1.

We define *APSyn* as the extent of the weighted intersection between the top-*N* most relevant contexts for the two words, where the weight is the average rank of the intersected features in the features lists of the target words, sorted by either PPMI or PLMI (note: other association measures can be also used).

Given a traditional count-based DSM, where every word is represented as a vector of weighted associations between such word and its contexts, this measure can be implemented through the following steps: i) first, for every target word in the pair, we rank the contexts according to the *Positive Pointwise Mutual Information* (PPMI: see 2.3.3.1); ii) second, once the contexts are ranked according to their PPMI, for every target word we pick the top-*N* contexts and we intersect them with those of the other word in the pair; iii) third, for each shared context, we add one divided by the average rank of the shared context in the two PPMI-ranked context lists. See the equation below:

$$APSyn(w_1, w_2) = \sum_{f \in N(F_1) \cap N(F_2)} \frac{1}{(rank_1(f) + rank_2(f))/2} \qquad 11$$

that is, for every feature *f* included in the intersection between the top-*N* features of w_1, *N(*F_1*)*, and w_2, *N(*F_2*)*, *APSyn* will add *1* divided the average rank of that feature in the two context lists, sorted by PPMI (note: other association measures can be also used). *N* is a parameter and $rank_x$ is the position of the intersected feature *f* in the context list of word *x*.

APSyn is expected to return the highest score when all top-*N* features are intersected and have exactly the same rank, as it happens in the identity. Lower scores are returned when the extent of the intersection or the relevance of the intersected contexts are smaller. Suppose, in fact, to have three toy-vectors *a*, *b* and *c*. They are initialized with the following features (note: between round brackets we report the indices of the dimensions and outside there are their values, which are used in *APSyn* only for ranking): *a = b = [2(1), 4(2), 3(3), 5(4), 1(5)]* and *c = [5(1), 2(2), 5(3), 1(4), 0(5)]*. If we want to calculate *APSyn* with *N=3* among *a* and *b*, and *b* and *c*, we need to:

a) sort the features and select only the top-*N=3*, obtaining: *a = b = [5(4), 4(2), 3(3)]* and *c = [5(1), 5(3), 2(2)]*.

b) calculate *APSyn* for the sorted vectors of *a* and *b*, obtaining:

$$APSyn(a,b) = \frac{1}{(1+1)/2} + \frac{1}{(2+2)/2} + \frac{1}{(3+3)/2} = 1 + 0.5 + 0.3 = 1.8$$

c) calculate *APSyn* for the sorted vectors of *b* and *c*, obtaining:

$$APSyn(b,c) = 0 + \frac{1}{(2+3)/2} + \frac{1}{(3+2)/2} = 0.4 + 0.4 = 0.8$$

As it can be seen the score of *APSyn* for *b* and *c* is much lower than the one for *a* and *b*, which are identities. Since it is based on the intersection, *APSyn* is not directional, so that *APSyn(b, c) = APSyn(c, b).*

3.3 Systematic Evaluation of APSyn vs. Vector Cosine

In the following sections, we report a systematic evaluation of *APSyn* on the most popular test sets – namely MEN (Bruni, Tran, & Baroni, 2013), WordSim-353 (Finkelstein, Gabrilovich, Matias, Rivlin, Solan, Wolfman, & Ruppin, 2001) and SimLex-999 (Hill, Roi, & Korhonen, 2014). In our evaluation, we compare the performance of our measure to the one of vector cosine under several parameter settings, taking into consideration corpus size, context window width, measure of association and the adoption of SVD. The results – measured in Spearman correlation – are also discussed in relation to the state-of-the-art VSMs, as they are reported in Hill, Roi, & Korhonen (2014). In particular, we compare our models (*vector cosine* is always calculated on the full vector, if not specified otherwise) to the neural language models (NLMs), which were identified as the best performing ones. In such comparison, *APSyn* and *vector cosine*, in their best settings, are competitive to, or even better than, word embeddings in almost all datasets (the only exception is WordSim-353 when the DSMs are trained on Wikipedia).

All the mentioned results for the NLMs are those calculated by Hill, Roi, & Korhonen (2014) using the code (or directly the embeddings) shared by the original authors. Collobert & Weston (2008)'s model was trained on 852 million words of Wikipedia and on the RCV Vol. 1 Corpus (Lewis, 2004). Huang, Manning, & Ng

(2012)'s model was trained on 990 million words of Wikipedia. The scores reported here for Mikolov, Yih, & Geoffrey (2013)'s model were obtained by Hill and colleagues training a 200 dimensions model, using Mikolov, Yih, & Geoffrey's *Word2Vec* software on 1000 million words of Wikipedia.

Finally, given the recent debate about the ability of DSMs to calculate genuine similarity as distinguished from word relatedness (see, for example, Hill, Roi, & Korhonen (2014)), we show how our models maximize such distinction performing particularly well in SimLex-999 and in the similarity subset extracted from WordSim-353 by Agirre, Alfonseca, Hall, Kravalova, Pasca, & Soroa (2009).

The remaining part of the chapter is organized as follows. Section 3.3.1 shortly describes the adopted corpora and the preprocessing. Section 3.3.2 illustrates the implementation of twenty-four DSMs. Section 3.3.3 describes the used datasets, while Sections 3.3.4 and 1.1.1 report and analyze the results of our experiments.

3.3.1 Corpora and Preprocessing

We used two different corpora for our experiments: RCV vol. 1 (Lewis, 2004) and the Wikipedia corpus (Baroni, Bernardini, Ferraresi, & Zanchetta, 2009), respectively containing 150 and 820 million words. These corpora are described in 2.3.1.

3.3.2 DSMs

We implemented twenty-four DSMs. All of them include the target words of the three datasets (i.e. MEN, WordSim-353 and SimLex-999) and the contexts with the frequency above 100. Other frequency thresholds were investigated (i.e. 250, 500 and 1000), but the models' performance was comparatively worst, and they were therefore abandoned.

We considered as contexts the content words (i.e. nouns, verbs and adjectives) within a window of 2, 3 and 5, even though the latter was given up for its poor performance. It might be relevant noticing here that targets and contexts are represented with their POS-tags in our DSMs. When the dataset does not explicitly mention the words' POS-tags (as it happens in WordSim-353), we assumed that both words had the same syntactic category, and we assigned it in the following way: *noun*, if both words exist as nouns, otherwise *verb,* if both words exist as verbs, otherwise *adjective*. This means that word pairs like [*white*, *rabbit*] and [*run*, *marathon*] are considered noun pairs.

Twelve out of twenty-four models were developed for RCV1, while other twelve were developed for Wikipedia. For each corpus, the twelve models differ according to the window size (i.e. 2 and 3), the adopted measure of association (i.e. frequency, PPMI and LMI) and the application of truncated SVD with *k=300* to the previous combinations. In the next sections, whenever we refer to a specific DSM, we will use the following convention: Corpus_SVD/NoSVD_AssociationMeasure_Window, such as in RCV_NoSVD_PPMI_2 or in Wiki_SVD_Freq_3.

3.3.3 Datasets

For our evaluation, we used three popular datasets: WordSim-353 (Finkelstein, Gabrilovich, Matias, Rivlin, Solan, Wolfman, & Ruppin, 2001), MEN (Bruni, Tran, & Baroni, 2013), SimLex-999 (Hill, Roi, & Korhonen, 2014). The three datasets have a different generative history, but all of them consist in word pairs with an associated score, that should either represent word association or word similarity.

WordSim-353 (Finkelstein, Gabrilovich, Matias, Rivlin, Solan, Wolfman, & Ruppin, 2001) was proposed as a word similarity dataset containing 353 pairs annotated with scores between 0 and 10. However, as claimed by Hill, Roi, & Korhonen (2014), the instructions given to the annotators were ambiguous with respect to similarity and association, so that i) many dissimilar word pairs received a high rating, and ii) only concepts that were dissimilar and not associated received low ratings. On top of it, WordSim-353 does not provide the POS-tags for the 439 words that it contains, forcing the users to decide which POS to assign to the ambiguous words (e.g. [*white*, *rabbit*] and [*run*, *marathon*]). An extension of this dataset is the subclassification carried out by Agirre, Alfonseca, Hall, Kravalova, Pasca, & Soroa (2009), who discriminated between similar and associated word pairs. Such discrimination was done by asking annotators to classify all pairs according to the semantic relation they hold (i.e. identical, synonymy, antonymy, hypernymy, meronymy and none-of-the-above). The annotation was then used to group the pairs in three categories: similar pairs (those classified as identical, synonyms, antonyms and hypernyms), associated pairs (those classified as meronyms and none-of-the-above, with an average similarity greater than 5), and non-associated pairs (those classified as none-of-the-above, with an average similarity below or equal to 5). Two gold standards were finally produced: i) one for similarity, containing 203 word pairs resulting from the union of similar and non-associated pairs; ii) one for relatedness, containing 252 word pairs resulting from the union of associated and non-associated pairs. Even though such classification made a clear distinction between the two types of relations (i.e. similarity and association), Hill, Roi, & Korhonen (2014) argue that these gold standards still carry the scores they had in WordSim-353, which are known to be ambiguous in this regard.

The MEN Test Collection (Bruni, Tran, & Baroni, 2013) includes 3,000 word pairs divided in two sets (one for training and one for testing) together with human judgments, obtained through Amazon Mechanical Turk. The 751 words composing the pairs were randomly selected from a list of words occurring at least 700 times in ukWac and Wackypedia corpora (size: about 2.7 billion tokens; Baroni, Bernardini, Ferraresi, & Zanchetta (2009): see 2.3.1) and at least 50 times as tags in the ESP game dataset (von Ahn and Dabbish, 2004). The construction was performed by asking subjects to rate which pair – among two – was the most related one (i.e. the most associated). Every pairs-couple was proposed only once, and a final score out of 50 was attributed to each pair, according to how many times it was rated as the most related. According to Hill, Roi, & Korhonen (2014), the major weakness of this dataset is that it does not encode word similarity, but a more general notion of association.

SimLex-999 is the dataset introduced by Hill, Roi, & Korhonen (2014) to address the above mentioned criticisms of confusion between similarity and association. The dataset consists of 999 pairs containing 1,028 words, which were also evaluated in terms of POS-tags and concreteness. The pairs were annotated with a score between 0 and 10, and the instructions strictly required the identification of word similarity, rather than word association. Hill and colleagues claim that differently from other datasets, SimLex-999 inter-annotator agreement has not been surpassed by any automatic approach.

3.3.4 Results

Given the twenty-four DSMs, for each dataset we have measured *APSyn* and *vector cosine* between the words in the pairs (Note: *APSyn* was not measured for the SVD models, because this measure was designed to measure the extent of intersection among linguistic contexts). Spearman correlation between our scores and the gold standard was then calculated for every model, as reported in Table 1 and Table 2. In particular, Table 1 describes the performances on SimLex-999, MEN and WordSim-353 for the RCV Vol. 1 models. Table 2, instead, describes the performances on the three datasets for the Wikipedia models. Concurrently, Table 3 and Table 4 describe the performances respectively of the RCV Vol. 1 and Wikipedia models on the subsets of WordSim-353 extracted by Agirre, Alfonseca, Hall, Kravalova, Pasca, & Soroa (2009).

Dataset		***SimLex-999***		***WordSim-353***		***MEN***	
Window		***2***	***3***	***2***	***3***	***2***	***3***
COS	*Cos Freq*	*0.149*	*0.133*	*0.172*	*0.148*	*0.089*	*0.096*
	Cos LMI	*0.248*	*0.259*	*0.321*	*0.32*	*0.336*	*0.364*
	Cos PMI	*0.284*	*0.267*	*0.41*	*0.407*	*0.424*	*0.433*
COS-SVD	*Cos SVD-Freq*	*0.128*	*0.127*	*0.169*	*0.173*	*0.076*	*0.084*
	Cos SVD-LMI	*0.19*	*0.21*	*0.299*	*0.291*	*0.275*	*0.286*
	Cos SVD-PMI	***0.386***	***0.382***	***0.485***	***0.470***	***0.509***	***0.538***
LMI-APSyn	*APSyn-1000*	*0.18*	*0.163*	*0.254*	*0.237*	*0.205*	*0.196*
	APSyn-500	*0.199*	*0.164*	*0.283*	*0.265*	*0.226*	*0.214*
	APSyn-100	*0.206*	*0.182*	*0.304*	*0.265*	*0.23*	*0.209*
PPMI-APSyn	*APSyn-1000*	*0.254*	*0.304*	*0.399*	*0.453*	*0.369*	*0.415*
	APSyn-500	*0.295*	*0.32*	***0.455***	***0.468***	*0.423*	*0.478*
	APSyn-100	***0.332***	***0.328***	*0.425*	*0.422*	***0.481***	***0.513***
STATE-OF-THE-ART							
Mikolov et al.		***0.282***		***0.442***		***0.433***	

Table 1: Spearman correlation scores for our twelve models trained on RCV Vol. 1, in the three datasets Simlex-999, WordSim-353 and MEN. At the bottom the performance of the state-of-the-art model of Mikolov, Yih, & Geoffrey (2013), as reported in Hill, Roi, & Korhonen (2014). The best performance for every model is bolded.

Dataset		***SimLex-999***		***WordSim-353***		***MEN***	
Window		**2**	**3**	**2**	**3**	**2**	**3**
COS	*Cos Freq*	*0.148*	*0.159*	*0.199*	*0.207*	*0.178*	*0.197*
	Cos LMI	*0.367*	*0.374*	*0.489*	*0.529*	*0.59*	*0.63*
	Cos PMI	*0.395*	*0.364*	*0.605*	*0.622*	*0.733*	*0.74*
COS-SVD	*Cos SVD-Freq*	*0.157*	*0.184*	*0.208*	*0.222*	*0.197*	*0.226*
	Cos SVD-LMI	*0.327*	*0.329*	*0.441*	*0.486*	*0.524*	*0.563*
	Cos SVD-PMI	***0.477***	***0.464***	***0.533***	***0.562***	***0.769***	***0.779***
LMI-APSyn	*APSyn-1000*	*0.343*	*0.344*	*0.449*	*0.477*	*0.586*	*0.597*
	APSyn-500	*0.339*	*0.342*	*0.438*	*0.470*	*0.58*	*0.588*
	APSyn-100	*0.303*	*0.31*	*0.392*	*0.428*	*0.48*	*0.498*
PPMI-APSyn	*APSyn-1000*	*0.434*	*0.419*	*0.599*	*0.643*	*0.749*	*0.772*
	APSyn-500	***0.442***	***0.423***	***0.602***	***0.653***	***0.757***	***0.773***
	APSyn-100	*0.316*	*0.281*	*0.58*	*0.608*	*0.703*	*0.722*
STATE-OF-THE-ART							
Huang et al.		*0.098*		*0.623*		*0.3*	
Collobert & Weston		*0.268*		*0.494*		*0.575*	
Mikolov et al.		***0.414***		***0.655***		***0.699***	

Table 2: Spearman correlation scores for our twelve models trained on Wikipedia, in the three datasets Simlex-999, WordSim-353 and MEN. At the bottom the performance of the state-of-the-art models of Collobert & Weston (2008), Huang, Manning, & Ng (2012), Mikolov, Yih, & Geoffrey (2013), as reported in Hill, Roi, & Korhonen (2014). The best performance for every model is bolded.

	WordSim (SIM)		**WordSim (REL)**	
	2	**3**	**2**	**3**
Cos Freq	0.208	0.158	0.167	0.175
Cos LMI	0.416	0.395	0.251	0.269
Cos PMI	0.52	**0.496**	0.378	**0.396**
Cos SVD-Freq	0.202	0.119	0.091	0.182
Cos SVD-LMI	0.39	0.368	0.18	0.189
Cos SVD-PMI	**0.574**	0.491	**0.408**	0.321
LMI APSyn-1000	0.32	0.290	0.259	0.241
LMI APSyn-500	0.355	0.319	0.261	0.284
LMI APSyn-100	0.388	0.335	0.233	0.270
PMI APSyn-1000	0.519	0.525	0.337	**0.397**
PMI APSyn-500	**0.564**	0.546	**0.361**	0.382
PMI APSyn-100	0.562	**0.553**	0.287	0.309

Table 3: Spearman correlation scores for our twelve models trained on RCV1, in the two subsets of WordSim-353. The best performance for every model is bolded.

	WordSim (SIM)		**WordSim (REL)**	
	2	**3**	**2**	**3**
Cos Freq	0.335	0.334	0.03	0.05
Cos LMI	0.638	0.663	0.293	0.34
Cos PMI	0.672	0.675	0.441	0.446
Cos SVD-Freq	0.35	0.363	-0.011	0.001
Cos SVD-LMI	0.6	0.626	0.223	0.286
Cos SVD-PMI	**0.722**	**0.725**	**0.444**	**0.486**
LMI APSyn-1000	0.609	0.609	0.317	0.36
LMI APSyn-500	0.599	0.601	0.289	0.344
LMI APSyn-100	0.566	0.574	0.215	0.271
PMI APSyn-1000	0.692	0.726	0.507	0.568
PMI APSyn-500	**0.699**	**0.742**	**0.508**	**0.571**
PMI APSyn-100	0.66	0.692	0.482	0.516

Table 4: Spearman correlation results for our twelve models trained on Wikipedia, in the subsets of WordSim-353. The best performance for every model is bolded.

3.3.5 Discussion

Table 1 shows the Spearman correlation scores for *APSyn* and vector cosine on the three datasets for the twelve DSMs built using RCV Vol. 1. Table 2 does the same for the DSMs built using Wikipedia. Vector cosine, as mentioned above, is calculated on the full vector, while *APSyn* is focused on the top-*N* contexts. For the sake of comparison, we also report the results of the state-of-the-art VSMs mentioned in Hill, Roi, & Korhonen (2014) and briefly described in Section 3.2.

Two models perform particularly well in comparison to the state-of-the-art: i) one relies on *APSyn*, applied on the PPMI weighted DSM (henceforth, *APSynPPMI*); ii) the other relies on the vector cosine applied on the SVD-reduced PPMI-weighted matrix (henceforth, *CosSVDPPMI*). These two models always outperform the state-of-the-art VSMs when RCV1 is used, and in SimLex-999 and MEN when Wikipedia is used (achieving competitive results also in WordSim-353).

Also, we can notice that the smaller window (i.e. 2) does not always perform better than the larger one (i.e. 3). However, this can be due to the minor difference between window 2 and window 3. In fact, despite Hill, Roi, & Korhonen (2014)'s claim that no evidence supports the hypothesis that smaller context windows improve the ability of models to capture similarity, we have noticed that window 5 was performing worse than the smaller ones, and therefore it was not further evaluated.

Some further observations are: i) the corpus size strongly affects the results; ii) PPMI strongly outperforms LMI in all models; iii) SVD boosts the vector cosine when it is combined with PPMI, while its contribution is unpredictable with other weighting measures. With reference to i), we should mention here that our version of Wikipedia is about one fifth smaller than the one used by Mikolov, Yih, & Geoffrey (2013), namely 820 million versus 1000 million words.

Looking more carefully at Table 1, we can see that our best models have a good advantage on the best state-of-the-art VSMs when RCV Vol. 1 is used. The advantage is reduced, yet existing, when Wikipedia is used as a training corpus (see Table 2). This may depend either on the difference between the versions of Wikipedia that were used to train the models, or on the ability of NLM to perform proportionally better with bigger amount of data. *APSyn* seems also to be affected by the corpus size: it, in fact, becomes more competitive with *vector cosine* model when used on Wikipedia.

Finally, few words need to be spent with regard to the ability of calculating genuine similarity, as opposed to word relatedness (see, for example, Turney (2001); Agirre, Alfonseca, Hall, Kravalova, Pasca, & Soroa (2009)). Table 3 and Table 4 show the Spearman correlation scores for the models respectively trained on RCV1 and Wikipedia in the subsets of WordSim-353 extracted by Agirre and colleagues (2009). It can be easily noticed that our best models work better on the similarity subset rather than on the relatedness one. In particular, they perform about 20-30% better better for similarity than for relatedness (see Table 3 and Table 4).

The good performances of our models on SimLex-999 (which was built with a particular attention to genuine similarity) and the large difference in performance between the two subsets of WordSim-353 described in Table 3 and Table 4 confirm that our models are indeed efficient in identifying genuine similarity. It is however possible that by adopting a larger window, the performance would have increased also for relatedness, as larger windows are expected to capture more topical relatedness than strictly semantic similarity. This trend can be already noticed in Table 4, where the scores for window 3 grow more on the related subset than on the similarity one. The fact that the trend is not instead visible in Table 3 may depend on the corpus size.

To summarize, *APSyn* is competitive with the best model of vector cosine (calculated on the full vector of a PPMI-SVD reduced matrix). The fact that the two measures implement different hypotheses (i.e. the former is rank-based while the latter is distance-based) may call for a combination of the two in order to merge their strength, possibly leading to a more complete approach to similarity. On top of it, we can already say here that *APSyn* is very scalable and its performance seems to grow with the size of corpus in higher proportion than what vector cosine does. In some recent experiments, we have noticed that when using large corpora, such as the concatenation of UkWac and Wacky (see 2.3.1), *APSyn* outperforms vector cosine calculated on the full vector extracted from the PPMI SVD-reduced matrix.

3.3.6 Feature Selection and Rank

In order to verify whether *vector cosine* calculated on the top-*N* features would also improve, we performed an experiment comparing *APSyn* to two new implementations of *vector cosine*, which calculate the distance of only the top-*N* features. The experiment was carried out on several DSMs built on a combination of ukWac and Wackypedia (see 2.3.1), containing about 2.7 billion words. We varied the window size, considering 2 and 5 content words on the left and the right of the targets. Only PPMI was used as weight, as in all previous experiments this has shown to perform more consistently. Two versions of Top-*N* vector cosine were implemented to calculate respectively i) the distance on the intersection between the top-*N* features (as it happens in *APSyn*); ii) the distance on the union between the top-*N* features. This second interpretation is a step ahead the hypothesis of *APSyn*. In fact, rather than evaluating the rank and the extent of the shared features, it measures how relevant the most salient feature of one vector are for the other. For consistency, we have also implemented a union version of *APSyn*. In Table 5 we report the results of the measures in the various settings.

As it can be seen in Table 5, vector cosine computed on the intersected features reports low results, which are even unpredictable when *N* is tuned. The reason is that the most relevant features of the vectors also have high scores and therefore a short distance. This causes *vector cosine* to score high for all the pairs, reducing therefore its discriminative power. Much more interesting results are obtained when vector cosine is calculated on the context union. In this case, it performs similarly to *APSyn*, without however never reach its Spearman scores.

In the union-based vector cosine, we have noticed that several contexts that are salient for one word are instead completely irrelevant for the other, therefore contributing very little or nothing to the vector cosine (i.e. their PPMI=0). This observation is also at the base of the slightly worst performance of APSyn for the union, compared to the more classic intersection.

From these experiments, we can conclude that context selection is certainly fundamental for *APSyn* performance. However, the results also demonstrate that the feature rank play a big role in the performance. We can conclude that the rank-based measure and the distance-based measure perform similarly, but catch two different aspects of word similarity. Combining them may therefore lead to further improvements.

Dataset		***SimLex-999***		***WordSim-353***		***MEN***	
Window		***2***	***5***	***2***	***5***	***2***	***5***
Vector Cosine PPMI	***N=All***	*0.268*	*0.227*	*0.603*	*0.584*	*0.729*	*0.707*
Vector Cosine PPMI SVD (k=300)	***N=All***	***0.301***	***0.268***	***0.618***	***0.617***	***0.73***	***0.713***
Vector Cosine Intersection	***100***	*0.219*	*0.181*	*0.383*	*0.384*	*0.487*	*0.483*
	500	*-0.075*	*-0.056*	*-0.184*	*-0.172*	*-0.217*	*-0.177*
	1000	*-0.125*	*-0.071*	*-0.252*	*-0.178*	*-0.236*	*-0.283*
Vector Cosine Union	***100***	***0.282***	***0.242***	***0.708***	***0.707***	***0.779***	***0.775***
	500	*0.275*	*0.242*	*0.699*	*0.702*	*0.768*	*0.758*
	1000	*0.274*	*0.241*	*0.684*	*0.681*	*0.761*	*0.751*
APSyn Intersection (Classic)	***100***	*0.277*	*0.208*	*0.520*	*0.552*	*0.668*	*0.665*
	500	*0.333*	*0.287*	*0.691*	*0.701*	*0.767*	*0.763*
	1000	***0.337***	***0.297***	***0.718***	***0.722***	***0.775***	***0.771***
APSyn Union	***100***	*0.325*	*0.278*	*0.704*	*0.703*	*0.779*	*0.773*
	500	*0.329*	*0.288*	*0.736*	*0.740*	*0.788*	*0.176*
	1000	***0.324***	***0.286***	***0.734***	***0.736***	***0.785***	***0.781***

Table 5: Spearman correlation scores for the three datasets Simlex-999, WordSim-353 and MEN, calculated with vector cosine computed on the full PPMI vector, the PPMI SVD reduced vector (k=300), the intersection of the top-*N* most relevant contexts and their union. Scores for APSyn computed on the intersection and union of the top-*N* contexts. The DSM are a 2- and 5-window based DSMs, with features weighted with PPMI. The best performance for every model is bolded.

3.4 Summary of Chapter III

In this chapter, we have illustrated the concept of similarity (Section 3.1), describing its properties. After the theoretical introduction, we have discussed the importance of similarity in NLP (Section 3.2) and we have illustrated several metrics that can be used for its identification, including our newly proposed *APSyn* (Section 3.2.1). In Section 3.3, we reported an extensive and systematic evaluation of the metrics in twenty-four count-based DSMs.

Each of the twenty-four models represents a particular configuration of parameters (i.e. corpus size, window size, weighting measure, dimensionality reduction, etc.) that have been carefully assessed. In particular, PPMI emerged as the most efficient association measure, especially when combined with SVD. The newly-introduced metric *APSyn* showed extremely promising performances, which encourage us to further investigations. *APSyn* is comparable to the best setting of vector cosine, and both of them (i.e. *APSynPPMI* and *CosSVDPPMI*) outperform the word embedding models in all datasets (with the exception of WordSim-353 for the models trained on Wikipedia, where competitive results are still obtained), independently of the training corpus, confirming Levy, Goldberg, & Dagan (2015)'s claim that well-tuned count-based DSMs can achieve similar performances to word embedding models, or even better ones. We concluded the discussion in 3.3.5 mentioning that *APSyn* seems to perform progressively better with bigger corpora, even outperforming the best setting of vector cosine when a large amount of data is used.

In Section 3.3.6, in order to verify whether vector cosine could improve by running on the top-*N* contexts, we implemented two versions of the vector cosine that calculate the distance respectively between: i) the intersection of the top-*N* features of the target words; and ii) the union of the top-*N* features of the target words. Results show that vector cosine on the intersection fails completely, while vector cosine on the union achieves competitive results, without however outperforming *APSyn*. We concluded therefore that the rank-based measure and the distance measure capture two different aspects of word similarity, which might be combined in future studies to achieve better results.

In the next chapter, we discuss the concept of opposition and introduce *APAnt*, our measure for its automatic identification. The measure is carefully evaluated in several antonym retrieval tasks.

Chapter IV – Opposition

"Truth, in its nature, is untruth."

M. Heidegger, *The Origin of the Work of Art*

This chapter describes the concept of semantic opposition (Section 4.1) and its treatment in NLP (Section 4.2), with a particular focus on distributional semantics. The second part of the chapter illustrates *APAnt* (Section 4.3), a newly introduced unsupervised method for the discrimination between antonyms and synonyms. *APAnt* was evaluated on numerous pairs extracted from EVALution, Lenci/Benotto and BLESS datasets, outperforming vector cosine and a baseline implementing the co-occurrence hypothesis (Section 4.4).

The chapter is an adaptation and extension of:

- Santus, E., Lu, Q., Lenci, A., & Huang, C.-R. (2014b). Taking Antonymy Mask off in Vector Space. Phuket, Thailand: Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation (PACLIC 2014).
- Santus, E., Lu, Q., Lenci, A., & Huang, C.-R. (2014c). Unsupervised Antonym-Synonym Discrimination in Vector Space. Pisa, Italy: Atti della Conferenza di Linguistica Computazionale Italiana (CLIC-IT 2014).

- Santus, E., Lenci, A., Lu, Q., & Huang, C.-H. (2015a). When Similarity Becomes Opposition: Synonyms and Antonyms Discrimination in DSMs. In *Italian Journal on Computational Linguistics*, aAccademia University Press.

4.1 Opposition

People do not always perfectly agree on classifying word pairs as opposites (Mohammad, Dorr, Hirst, & Turney, 2013), confirming that their identification is indeed a hard task, even for native speakers. The major problems in such task are that i) opposites are rarely in a truly binary contrast (e.g. *warm*/*hot*); ii) the contrast can be of different kinds (e.g. semantic, as in *hot/cold*, or referential, as in *Clinton/Bush*); and iii) opposition is often context-dependent (e.g. consider the near-synonyms *very good* and *excellent* in the following sentence: "not simply *very good*, but *excellent*"; Cruse (1986)). All these issues make opposites difficult to define, so that linguists often need to rely on diagnostic tests to make the opposition clear (Murphy M. L., 2003).

Over the years, many scholars from different disciplines have tried to provide a precise definition of this semantic relation. They are yet to reach any conclusive agreement. Kempson (1977) defines opposites as word pairs with a "binary incompatible relation", such that the presence of one meaning entails the absence of the other. In this sense, *giant* and *dwarf* are good opposites, while *giant* and *person* are not. Mohammad, Dorr, Hirst, & Turney (2013), noticing that the terms *opposites*, *contrasting* and *antonyms* have often been used interchangeably, have proposed the following distinction: i) *opposites* are word pairs that are strongly incompatible with each other and/or are saliently different across a dimension of meaning; ii) *contrasting word pairs* have some non-zero degree of binary incompatibility and/or some non-zero difference across a dimension of meaning; iii) *antonyms* are opposites that are also gradable adjectives. They have also provided a simple but comprehensive classification of opposites based on Cruse (1986), including a) *antipodals* (e.g. *top-bottom*), pairs whose terms are at the opposite extremes of a specific meaning dimension; b) *complementaries* (e.g. *open-shut*), pairs whose terms divide the domain in two mutual exclusive compartments; c) *disjoints* (e.g. *hot-cold*), pairs whose words occupy non-overlapping regions in a specific semantic dimension, generally representing a state; d) *gradable opposites* (e.g. *long-short*), adjective- or adverb-pairs that describe gradual semantic dimensions, such as length, speed, etc.; e) *reversibles* (e.g. *rise-fall*), verb-pairs whose words respectively describe the change from state A to state B and the inverse, from state B to state A.

In this chapter, we will not account for all these differences, but rather we will use the terms *opposites* and *antonyms* as synonyms, meaning all pairs of words in which a certain level of contrast is perceived. Under such category we include also the *paranyms*, which are a specific type of coordinates (Huang, Su, Hsiao, & Ke, 2007) that partition a conceptual field into complementary subfields. For instance, although *dry season*, *spring*, *summer*, *autumn* and *winter* are all co-hyponyms, only the latter four are paranyms, as they split the conceptual field of *seasons*.

According to Cruse (1986), antonymy is characterized by the *paradox of simultaneous similarity and difference*: opposites are identical in every dimension of meaning except for one. A typical example of such paradox is the relation between *dwarf* and *giant*. These words are semantically similar in many aspects (i.e. they may refer to similar entities, such as *humans*, *trees*, *galaxies*), differing only for what concerns the size, which is assumed to be a salient semantic dimension for them. From a distributional perspective, *dwarfs* and *giants* share many contexts (e.g., both *giant* and *dwarf* may be used to refer to *galaxies*, *stars*, *planets*, *companies*, *people*[9]), differing for those related to the semantic dimension of size. For example, *giant* is likely to occur in contexts related to big sizes, such as *global*, *corporate*, *dominate* and so on[10], while *dwarf* is likely to occur in contexts related to small sizes, such as *virus*, *elf*, *shrub* and so on[11].

4.2 Antonymy in NLP

Opposites identification is very challenging for computational models (Mohammad, Dorr, & Hirst, 2008). Yet, this relation is essential for many NLP applications, such as *Information Retrieval* (IR), *Ontology Learning* (OL), *Machine Translation* (MT), *Sentiment Analysis* (SA) and *Dialogue Systems* (Roth & Schulte im Walde, 2014; Mohammad, Dorr, Hirst, & Turney, 2013). In particular, the automatic identification of semantic opposition is crucial for the detection and generation of paraphrases (i.e. during the generation, similar but contrasting candidates should be filtered out, as described in Marton (2011)), the understanding

[9] These examples were found through the *Sketch Engine* (https://www.sketchengine.co.uk), by using the *word sketch* function.

[10] Ibid.

[11] Ibid.

of contradictions (de Marneffe, Rafferty, & Manning, 2008) and the identification of irony (Xu, Santus, Laszlo, & Huang, 2015) and humor (Mihalcea, 2005).

Several existing hand-crafted computational lexicons and thesauri explicitly encoding opposition are often used to support the above mentioned NLP tasks, even though many scholars have shown their limitations. Mohammad, Dorr, Hirst, & Turney (2013), for example, point out that "more than 90% of the contrasting pairs in GRE closest-to-opposite questions[12] are not listed as opposites in WordNet". Moreover, the relations encoded in such resources are mostly context independent.

Most of corpus-based approaches on opposition are founded on the *co-occurrence hypothesis* (Lobanova, 2012), formulated by Miller & Charles (1991) after observing that opposites co-occur in the same sentence more often than expected by chance. Such claim has then found many empirical confirmations (Justeson & Katz, 1991; Fellbaum C. , 1995) and it is used in the present work as a baseline. Ding & Huang (2013) also pointed out that, unlike co-hyponyms, opposites generally have a strongly preferred word order when they co-occur in a coordinate context (i.e. A and/or B), such as in *dead or alive*. Another part of related research has been focused on the study of lexical-syntactic constructions that can work as linguistic tests for opposition definition and classification (Cruse, 1986).

Starting from all these observations, several computational methods for opposition identification were implemented. Most of them rely on patterns (Lobanova, Kleij, & Spenader, 2010; Turney, 2008; Pennacchiotti & Pantel, 2006),

[12] GRE stands for *Graduate Record Examination*, which is a standardized test, often used as an admissions requirement for graduate schools in the United States.

which are known to be affected by various problems, most notably the difficulty of finding patterns that are highly reliable and uniquely associated with specific relations, without incurring at the same time in data-sparsity problems. The experience with pattern-based approaches has shown that these two criteria can rarely be satisfied simultaneously. Some other methods, like the one proposed by Lucerto, Pinto, & Jiménez-Salazar (2002), use the number of tokens between the target words and other clues (e.g. the presence/absence of conjunctions like *but*, *from*, *and*, etc.) to identify contrasting words.

Turney (2008) proposed a supervised algorithm for the identification of several semantic relations, including synonyms and opposites. The algorithm relied on a training set of word pairs with class labels to assign the labels also to a testing set of word pairs. All word pairs were represented as vectors encoding the frequencies of co-occurrence in textual patterns extracted from a large corpus of web pages. He used the Sequential Minimal Optimization (SMO) Support Vector Machine (SVM) with a radial basis function kernel implemented in Weka (Waikato Environment for Knowledge Analysis; see Witten, Frank, & Hall (2005)). In the discrimination between synonyms and opposites, the system achieved an accuracy of 75% against a majority class baseline of 65.4%.

Mohammad, Dorr, & Hirst (2008) proposed a method for determining the degree of semantic contrast based on the use of thesauri categories and corpus statistics. For each target word pair, they used the co-occurrence and the distributional hypothesis to establish the degree of opposition. Their algorithm achieved an F-score of 0.7, against a random baseline of 0.2.

Mohammad, Dorr, Hirst, & Turney (2013) used an analogical method based on a given set of contrasting words to identify and classify different kinds of opposites by hypothesizing that for every opposing pair of words, A and B, there is at least another opposing pair, C and D, such that A is similar to C and B is similar to D. For example, for the pair *night-day*, there is the pair *darkness-daylight*, such that *night* is similar to *darkness* and *day* to *daylight*. Given the existence of contrast, they calculated its degree relying on the *co-occurrence* hypothesis. Their approach outperformed other state-of-the-art measures.

Schulte im Walde & Köper (2013) proposed a vector space model relying on lexico-syntactic patterns to distinguish between synonymy, antonymy and hypernymy. Their approach was tested on German nouns, verbs and adjectives, achieving a precision of 59.80%, which was above the majority baseline.

More recently, Roth & Schulte im Walde (2014) proposed that statistics over discourse relations can be used as indicators for paradigmatic relations, including opposition.

4.3 APAnt: Discrimination of Antonyms and Synonyms

Starting from the *paradox of simultaneous similarity and difference between antonyms* (Cruse, 1986), we propose a rank-based distributional measure inspired at the *Average Precision* formula (see: Kotlerman, Dagan, Szpektor, & Zhitomirsky-Geffet (2010)) to discriminate antonyms from near-synonyms, under the assumption that both have similar distributions but antonyms share a smaller proportion of their most relevant contexts. For example, *dress* and *clothe* are very likely to have among their most relevant contexts words like *wear, thick, light* and so on. On the other hand, *dwarf* and *giant* will probably share contexts like *eat* and *sleep*, but they will differ on other very salient contexts such as *big* and *small*. To exemplify such idea, in Table 6 we report the first most relevant contexts for the verbs *to fall*, *to lower* and *to raise*, where the latter are respectively near-synonym and antonym of *to fall*.

APAnt takes into account two main factors: i) the extent of the intersection between the top-*N* most relevant contexts of two words (where relevance is measured as the rank in a LMI-ranked contexts list); and ii) the salience of such intersection (i.e. the average rank of the context in the two targets LMI-ranked contexts lists). It can be seen as the inverse of *APSyn*. Such an inverse should not be confused with the inverse of vector cosine. In fact, while the inverse of vector cosine is a measure of dissimilarity (i.e. words having different distribution), the inverse of *APSyn* – i.e. *APAnt* – simply measures how different the most relevant contexts are. We expect near-synonyms to share many relevant contexts (i.e. scoring high with *vector cosine* and *APSyn*, and consequently low for *APAnt*), while antonyms are expected to share many contexts, but a lower proportion of their most relevant ones (i.e. scoring high with *vector cosine* and low with *APSyn*, and consequently high for *APAnt*).

As we have described it in Section 3.2.1, given a target pair $w_1$and w_2, *APSyn* first selects the *N* most relevant contexts for each of the two terms and, then, calculates the extent of their intersection, by summing up for each intersected context a function of its salience score. *N* should be large enough to sufficiently describe the distributional semantics of a term for a given purpose (i.e. in our experiments we have often chosen values like 50, 100, 150, 200 and 250, but this parameter can be further optimized). It is important to note here that a small *N* would inevitably reduce the intersection, forcing most of the scores to the same values (and eventually to zero), independently on the relation the pair under examination holds. On the other hand, a very large value of *N* will inevitably include also contexts with very low values of LMI and, therefore, much less relevant for the target pairs. Finally, it might be relevant to notice that *APSyn* assigns the highest scores to the identity pairs, as their intersection will include all *N* contexts and all in the same rank (e.g. *dog-dog*). Its inverse, instead, would do exactly the opposite. While *APSyn* assigns higher scores to near-synonyms, *APAnt* assigns higher scores to antonyms. Such scores can then be used for semantic relations discrimination tasks:

$$APAnt(w_1, w_2) = \frac{1}{APSyn(w_1, w_2)}$$

where APSyn is defined as in 3.2.1.

To fall	To lower (Synonym)	To raise (Antonym)
1. love-n	1. cholesterol-n	1. awareness-n
2. category-n	2. raise-v	2. fund-n
3. short-j	3. level-n	3. money-n
4. disrepair-n	4. blood-n	4. issue-n
5. rain-n	5. cost-n	5. question-n
6. victim-n	6. pressure-n	6. concern-n
7. price-n (rank=7)	**7. rate-n** (rank=7)	7. profile-n
8. disuse-n	**8. price-n** (rank=8)	8. bear-v
9. cent-n	9. risk-n	9. standard-n
10. rise-v	10. temperature-n	10. charity-n
11. foul-j	11. water-n	11. help-v
12. hand-n	12. threshold-n	12. eyebrow-n
13. trap-n	13. standard-n	13. level-n
14. snow-n	14. flag-n	14. aim-v
15. ground-n	15. age-n	15. point-n
16. rate-n (rank=16)	16. lipid-n	16. objection-n
17. …	17. …	17. …

Table 6: Top 16 contexts for the verbs *to fall*, *to lower* and *to raise*. These terms are present in our dataset. At this cutoff, the antonyms do not yet share any context.

Two cases need to be considered here:

- if *APSyn* has not found any intersection among the *N* most relevant contexts, it will be set to zero, and consequently *APAnt* will be infinite;
- if *APSyn* has found a large and salient intersection, it will get a high value, and consequently the value of *APAnt* will be very low.

The first case happens when the two terms in the pair are distributionally unrelated or when *N* is not sufficiently high. Therefore, *APAnt* is set to the maximum attested value. The second case, instead, can occur when two terms are distributionally very similar, sharing therefore many salient contexts. Ideally, this should only be the case for near-synonyms.

As we will see in Section 4.4.2, most of the scores given by *APAnt* are either very high or very low. In order to scale them, in Section 4.4.2 we use the logarithm function of the scores, while in Section 4.4.3 we normalize them through the *Min-Max function* (note: our infinite values will be set – together with the maximum ones – to the highest attested finite value):

$$MinMax(x_i) = \frac{x_i - \min(X)}{\max(X) - \min(X)} \qquad 12$$

where, for every value x_i that we want to normalize, we calculate the ratio between x_i less the minimum score in the distribution and the difference between the maximum and the minimum score in the distribution (i.e. the score variability range).

4.4 Evaluation

In this section we report the main experiments we carried out to evaluate *APAnt*. Section 4.4.2 summarizes the experiments reported in Santus, Lu, Lenci, & Huang (2014c), showing the box-plots (which describe the distributions of scores per relation) and reporting the *Average Precision* measure (AP; see Kotlerman, Dagan, Szpektor, & Zhitomirsky-Geffet (2010)), which is used to compute the ability of *APAnt* to discriminate antonyms from synonyms for nouns, adjectives and verbs. For comparison, we report the performances of the vector cosine and of a baseline implementing the co-occurrence hypothesis. Section 4.4.3 reports the experiments described in Santus, Lenci, Lu, & Huang (2015a), where *APAnt* is tested on a larger dataset and it is not only used to discriminate antonyms from synonyms, but also from hypernyms and co-hyponyms. Interestingly, in these experiments we found that *APAnt* obtains high scores for hypernyms too.

4.4.1 DSM and Datasets

For our experiments, we use a standard window-based DSM recording co-occurrences with context window of the nearest two content words both to the left and right of each target word. Co-occurrences are extracted from a combination of the freely available ukWaC and WaCkypedia corpora (with a respective size of 1.915 billion and 820 million words). See Section 2.3.1 for further details about the corpora.

For the experiments described in 4.4.2, we rely on a subset of English word pairs collected by Alessandro Lenci and Giulia Benotto in 2012/13 using Amazon Mechanical Turk (Benotto, 2015), following the method described by Scheible & Schulte im Walde (2014). The balance of the target items across word categories, their frequency, the degree of ambiguity and the semantic classes were some of the criteria adopted for collecting the data. Our subset contains 2,232 word pairs[13], including 1,070 antonym pairs and 1,162 synonym pairs. Antonyms include 434 noun pairs (e.g. *parody-reality*), 262 adjective pairs (e.g. *unknown-famous*) and 374 verb pairs (e.g. *try-procrastinate*). Synonyms include 409 noun pairs (e.g. *completeness-entirety*), 364 adjective pairs (e.g. *determined-focused*) and 389 verb pairs (e.g. *picture-illustrate*).

[13] The sub-set includes all the pairs for which both the target words exist in the DSM.

For the experiments described in 4.4.3, the datasets change according to the two subtasks that we performed. The first subtask is concerned with the discrimination between synonyms and antonyms, while the second one introduces also hypernyms and co-hyponyms. Both the datasets are extracted from the English word pairs obtained with the union of the Lenci/Benotto dataset (Benotto, 2015), BLESS (Baroni & Lenci, 2011) and EVALution 1.0 (Santus, Yung, Lenci, & Huang, 2015b). The final dataset for task 1 contains 4,735 word pairs, including 2,545 antonyms and 2,190 synonyms. The class of antonyms consists of 1,427 noun pairs (e.g. *parody-reality*), 420 adjective pairs (e.g. *unknown-famous*) and 698 verb pairs (e.g. *try-procrastinate*). The class of synonyms consists of 1,243 noun pairs (e.g. *completeness-entirety*), 397 adjective pairs (e.g. *determined-focused*) and 550 verb pairs (e.g. *picture-illustrate*). The final dataset for task 2, includes also 4,261 hypernyms from the Lenci/Benotto dataset, BLESS and EVALution, and 3,231 coordinates from BLESS. The class of hypernyms consists of 3,251 noun pairs (e.g. *violin-instrument*), 364 adjective pairs (e.g. *able-capable*) and 646 verb pairs (e.g. *journey-move*). The coordinates only include noun pairs (e.g. *violin-piano*).

4.4.2 Experiments with Lenci/Benotto

BOX-PLOTS. Box-plots display the median of a distribution as a horizontal line within a box extending from the first to the third quartile, with whiskers covering 1.5 of the interquartile range in each direction from the box, and outliers plotted as circles.

Figure 2 and Figure 3 show the box-plots summarizing the logarithmic distributions of *APAnt* and baseline scores for antonyms and synonyms, respectively. The logarithmic distribution is used to smooth the range of data, which would otherwise be too large and sparse for the box-plot representation. Figure 4 shows the box-plot summarizing the vector cosine scores. Since vector cosine scores range between 0 and 1, we multiplied them by ten to scale up for comparison with the other two box-plots in Figure 2 and Figure 3. The box-plots in Figure 2, Figure 3 and Figure 4 include test data with all part of speech types (i.e. nouns, adjectives and verbs). The box-plots for individual parts-of-speech are not reported because they do not show significant differences.

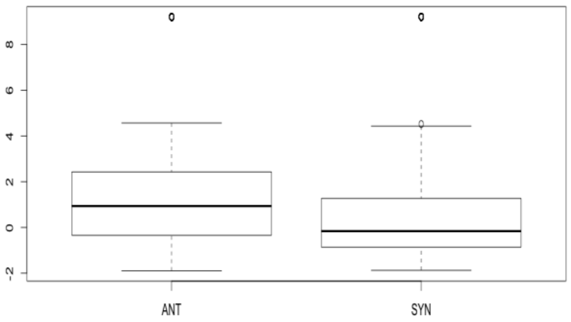

Figure 2: Logarithmic distribution of *APAnt* scores for antonym and synonym pairs (*N*=100) across nouns, adjectives and verbs.

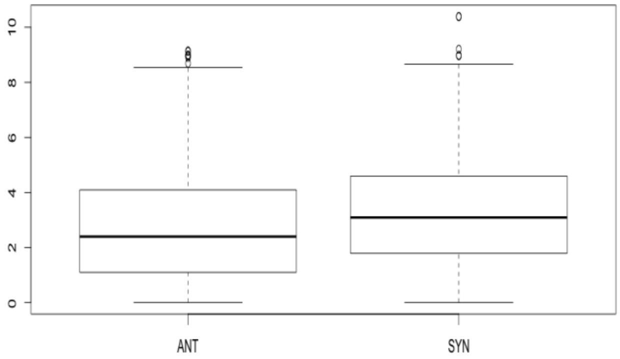

Figure 3: Logarithmic distribution of the co-occurrence baseline scores for antonym and synonym pairs across nouns, adjectives and verbs[14].

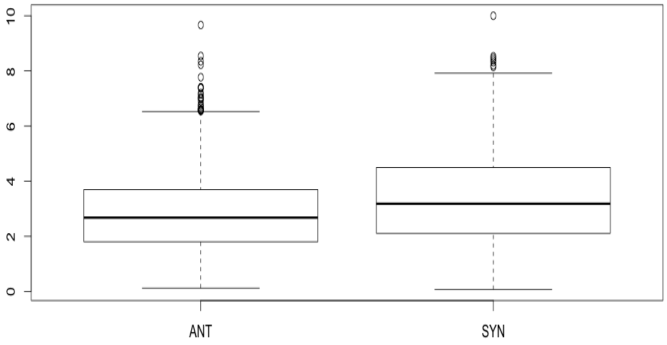

Figure 4: Distribution of the vector cosine scores for antonym and synonym pairs across nouns, adjectives and verbs[15].

The more the boxes in the plot overlap, the less distinctive the measure is. In Figure 3 and Figure 4, we can observe that the baseline and the *vector cosine* tend to

[14] 410 pairs with co-occurrence equal to zero on a total of 2,232 have been removed to make the box-plot readable, because *log(0) = -inf*

[15] Since *vector cosine* scores range between 0 and 1, we multiplied them by ten to scale up for comparison with the other two box-plots in Figure 2 and Figure 3**Error! Main Document Only.**.

promote synonyms on antonyms. Also, we can observe that there is a large range of overlap among synonyms and antonyms distributions, showing the weakness of these two measures for their discrimination. On the other hand, in Figure 2 we can observe that *APAnt* scores are much higher for antonymy-related pairs. This shows that *APAnt* has a clear preference for antonyms, differently from the vector cosine or the simple co-occurrence. Moreover, results also suggest the partial inaccuracy of the co-occurrence hypothesis. The tendency of co-occurring is not a hallmark of antonyms, being a property of near-synonyms too.

AVERAGE PRECISION. Table 7 shows the second performance measure we used in our evaluation: Average Precision (AP; see: Kotlerman, Dagan, Szpektor, & Zhitomirsky-Geffet (2010)) computed for *APAnt*, baseline and vector cosine scores. AP is used in Information Retrieval to combine precision, relevance ranking and overall recall. It corresponds to the area under the precision-recall curve and it is defined by the following equation:

$$AP = \sum_{k=1}^{M} p(k)\Delta r(k)$$

where: *k* varies between 1 and the total number of elements in the rank (i.e. *M*); *p(k)* is the precision at cutoff *k* (i.e. how many relevant elements were identified among the total elements present at cutoff *k*); and $\Delta r(k)$ is the change in recall that happens between cutoff *k-1* and cutoff *k* (i.e. recall increases if a relevant element is in position *k*). For example, the AP of a vector like *a = [1 (T), 1 (T), 0 (F), 0 (T), 1 (T)]* (where 1 is the prediction for (T)rue and 0 is the prediction for (F)alse; between round brackets we report the gold standard values) would be the following sum:

$1 * 0.25 + 1 * 0.25 + 0.66 * 0 + 0.5 * 0.25 + 0.66 * 0.25 = 0.79$. The best possible score we can obtain is 1 for antonymy and 0 for synonymy, which would correspond to the perfect discrimination between antonyms and synonyms.

APAnt performs the best, compared to the reference methods, which mostly promote synonyms on antonyms. In fact, regardless of the value of *N* (either equal to 50, 100 or 150), the AP scores confirm the trend shown in the box-plots of Figure 2, Figure 3 and Figure 4, proving that *APAnt* is a very effective measure to distinguish antonymy from synonymy.

ALL PoS	**ANT**	**SYN**
APAnt, N=50	0.71	0.57
APAnt, N=100	**0.73**	**0.55**
APAnt, N= 150	0.72	0.55
Baseline	0.56	0.74
Cosine	0.55	0.75

Table 7: Average Precision (AP) values per relation for *APAnt* (*N=50*, *100* and *150*), baseline and vector cosine across the parts-of-speech.

Below we also list the AP values for the different parts-of-speech (i.e. nouns, adjectives and verbs) with the parameter *N=100*.

NOUNS	ANT-N	SYN-N
APAnt, N=100	**0.79**	**0.48**
Baseline	0.53	0.77
Cosine	0.54	0.74

Table 8: Average Precision (AP) values per relation for *APAnt*, baseline and vector cosine on nouns.

ADJECTIVES	ANT-J	SYN-J
APAnt, N=100	**0.65**	**0.65**
Baseline	0.57	0.74
Cosine	0.58	0.73

Table 9: Average Precision (AP) values per relation for *APAnt*, baseline and vector cosine on adjectives.

VERBS	ANT-V	SYN-V
APAnt, N=100	**0.74**	**0.52**
Baseline	0.53	0.75
Cosine	0.52	0.77

Table 10: Average Precision (AP) values per relation for *APAnt*, baseline and vector cosine on verbs.

As it can be observed, *APAnt* always outperforms the baseline. However, a lower performance can be noticed in Table 9, where the AP scores for adjectives are 0.65 for both antonyms and synonyms. A possible explanation of this result might be that the different number of pairs per relation influences the AP values. In our dataset, in fact, we have 364 synonymy-related pairs against 262 antonym pairs for adjectives (+102 synonymy-related pairs, +39%).

ADJECTIVES	**ANT-J**	**SYN-J**
APAnt, N=100	**0.72**	**0.6**
Baseline	0.66	0.69
Cosine	0.68	0.66

Table 11: Average Precision (AP) values per relation for *APAnt*, baseline and vector cosine on adjectives, after extracting 262 pairs per relation.

To test this hypothesis, we randomly extract 262 synonymy-related pairs from the 364 that are present in our dataset and we re-calculate the AP scores for both the relations. The results can be found in Table 11. Such results confirm that *APAnt* works properly also for adjectives. However, this is the lowest result among the three parts-of-speech used in our experiments.

The different results for the three parts-of-speech should be interpreted in relation to our hypothesis. It is in fact possible that while opposing nouns (e.g. *giant – dwarf*) share very few or no salient contexts, opposing verbs (e.g. *rise – fall*) and –

even more – opposing adjectives (e.g. *hot – cold*) share some salient contexts, making the discrimination task more difficult for these parts-of-speech. In any case, the accuracy of our method has strongly outperformed the baseline for all the parts-of-speech, confirming the robustness of our hypothesis.

4.4.3 Experiments with Joint Datasets

In Table 12, we report the AP values for *APAnt* and the baselines. Since the Average Precision values may be biased by pairs obtaining the same scores (in these cases, in fact, the rank cannot be univocally determined, except by assigning it randomly or adding a new criterion – and we have adopted the alphabetic one), for every measure, we provide information about how many pairs have identical scores. As it can be seen in the table, when *N* is big enough (in our case *N>=200*), *APAnt* has less identical scores than the *vector cosine*.

As it can be seen in Table 12, *APAnt* outperforms all the baselines. Given that our dataset contains few more antonyms than synonyms, we expect the *random rank* to have a certain preference for antonyms. This is, in fact, what happens, making the random baseline outperforming the *co-occurrence baseline*. The vector cosine, instead, has a preference for synonyms, balancing the AP independently of the different sizes of the two classes. Finally, we can notice that while the values of *N* seem to have a small impact on the performance, they have a high impact in reducing the number of identical scores. That is, the larger the value of *N*, the less pairs have identical scores. Co-occurrence frequency is the worst measure in this sense, since almost 76% of the pairs obtained identical scores. Such a high value has to be attributed to the sparseness of the data and may be eventually reduced by choosing a larger window in the construction of the DSM. However, this also shows that use of co-occurrence data alone may be of little help in discriminating antonyms from other semantic relations.

MEASURE	N (Pairs with identical score)	Antonyms	Synonyms
APAnt	50 (1374)	0.60	0.41
APAnt	100 (274)	0.60	0.41
APAnt	150 (96)	0.61	0.41
APAnt	200 (67)	0.61	0.40
APAnt	250 (67)	0.61	0.40
Co-occurrence	(3591)	0.54	0.46
Cosine	(85)	0.5	0.5
Random	(3)	0.55	0.45

Table 12: AP scores for *APAnt* and the baselines on the dataset containing 4,735 word pairs, including 2,545 antonyms and 2,190 synonyms. The second column contains the values of *N* (only for *APAnt*) and – between brackets – the quantity of pairs having identical scores.

In Table 13, we report the AP scores for the task performed on the dataset including also hypernyms and coordinates. Again, *APAnt* outperforms the baselines. An interesting and unexpected result is obtained for the hypernyms. Even though their class is almost twice the size of antonyms and synonyms (this can be seen also in the AP scores obtained by the baselines), this result is important and is further discussed below. Once more, the AP value for the *random rank* is proportional to the sizes of the classes. Co-occurrence frequency seems to have a slight preference for antonyms and hypernyms (which may be due to the size of these classes), while the vector cosine seems to prefer synonyms and coordinates.

Once more, the values of *N* do not significantly affect the AP scores, but they influence the number of identical scores (*N>=150* is necessary to have less identical scores than those obtained with the vector cosine). Co-occurrence frequency is again the worst measure in this sense, since it has as many as 10,760 pairs with the same score on 12,227 (88%).

MEASURE	**N (Pairs with identical score)**	**Antonyms**	**Synonyms**	**Hypernyms**	**Coordinates**
APAnt	50 (4756)	0.27	0.18	0.43	0.18
APAnt	100 (2449)	0.27	0.18	0.44	0.17
APAnt	150 (1987)	0.28	0.18	0.44	0.17
APAnt	200 (1939)	0.28	0.18	0.44	0.17
APAnt	250 (1901)	0.28	0.18	0.44	0.17
Co-occ.	(10760)	0.23	0.19	0.36	0.23
Cosine	(2096)	0.2	0.2	0.31	0.29
Random	(15)	0.21	0.18	0.35	0.26

Table 13: AP scores for the *APAnt* and the baselines on the dataset containing 12,227 word pairs, including 4,261 hypernyms and 3,231 coordinates. The second column contains the values of *N* (only for *APAnt*) and – between brackets – the quantity of pairs having identical scores.

The AP results confirm that *APAnt* assigns higher scores to antonyms compared to both synonyms and coordinates. Such results are coherent with our hypothesis that antonyms share less relevant contexts than both synonyms and coordinates. Figure 5 shows boxplots describing the distribution of scores for *APAnt* (on the left) and *vector cosine* (on the right). As it can be seen, *APAnt* scores are – on average – higher for antonymy, while the vector cosine scores are similarly distributed for both relations, with a slight preference for near-synonyms.

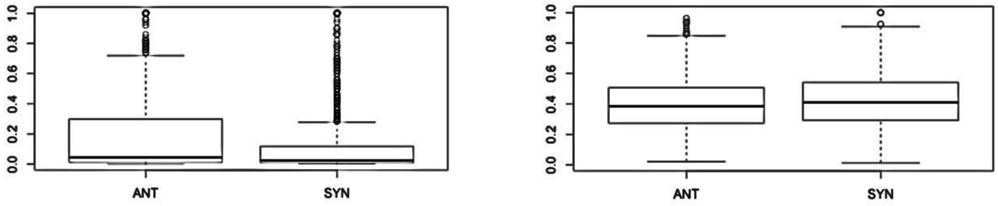

Figure 5: APAnt scores (on the left) for N=50 and *vector cosine* ones (on the right). APAnt scores have been normalized in a range 0-1 with Min-Max (see: Section 4.3), setting the infinite values to the maximum attested value, and therefore normalized to 1.

A surprising result instead occurs for the class of hypernyms, as shown in Table 13, to which *APAnt* assigns high scores. Although such class is almost twice the size of both antonyms and synonyms, the *APAnt* AP score for such class is much higher than the AP scores assigned to the baselines. The reason may be that hypernymy related pairs – even though they are known to be characterized by high distributional similarity – do not share many salient contexts. That is, contexts that are salient for one of the two terms are not necessarily salient for the other one (e.g. *wild* for the hypernym *animal* might not be salient for the hyponym *dog*), and *vice versa* (e.g. *bark* is not salient for the hypernym *animal*, while it is for the hyponym *dog*). This result is coherent with what we have found in Santus, Lenci, Lu, & Schulte im Walde (2014a) (see Section 5.3), where we have shown that the most relevant contexts of hypernyms tend to have higher entropy than the most relevant contexts of hyponyms. More investigation is required in this respect, but it is possible that *APAnt* can be used in combination with other measures (e.g. *SLQS* or entropy: see Section 5.3) for discriminating also hypernymy.

Another relevant point is the role of *N*. As it can be seen from the results, it has a low impact on the AP values, meaning that the rank is not strongly affected by its change (at least for what concerns the values we have tested, which are 50, 100, 150, 200 and 250). However, the best results are generally obtained with *N>150*. The value of *N* is instead inversely proportional to the number of identical scores.

Finally, we have identified a potential bias for AP, concerned with the ranking of pairs that have obtained the same score. In our experiment, we have used the alphabetical order as the secondary *criterion* for ranking. Such criterion does not affect the evaluation of APAnt (including its variants) and vector cosine, because these measures assign a fairly small amount of identical scores (around 15% of 12,227 pairs). It instead certainly affects the reliability of the co-occurrence frequency, where the amount of pairs obtaining identical scores amount up to 88%. Even though such result is certainly imputable to the sparseness of the data, we should consider whether the co-occurrence frequency can properly account for antonymy.

4.5 Summary of Chapter IV

This chapter has presented *APAnt* (Section 4.3), a distributional measure for the identification of antonymy based on a distributional interpretation of the *paradox of simultaneous similarity and difference between the antonyms* (Cruse, 1986).

APAnt is evaluated in several discrimination tasks (Section 4.4), including synonyms, antonyms, hypernyms and coordinates. The evaluation has been performed on nouns, adjectives and verbs. In the tasks, *APAnt* has outperformed the vector cosine and the baseline implementing the co-occurrence hypothesis (Fellbaum C. , 1995; Justeson & Katz, 1991; Miller & Charles, 1991) for all the parts-of-speech, achieving good AP for all of them. However, its performance is higher for nouns, slightly lower for verbs and significantly lower for adjectives. These differences across parts-of-speech might be due to the fact that while opposing nouns share very few salient contexts, opposing verbs and – even more – opposing adjectives share some salient contexts, making the discrimination task more difficult. *APAnt* performance supports our hypothesis, according to which synonyms share a number of salient contexts that is significantly higher than the one shared by antonyms.

The chapter has also discussed a limitation of AP, when the measures assign too many identical scores. In this respect, *APAnt* outperforms the vector cosine when *N>150*, producing less similar scores than it.

Finally, unexpectedly, *APAnt* has been found to have preferences also for hypernymy, which redirect us to the hypothesis presented in the next chapter, according to which the most relevant contexts of hypernyms are less informative than those of hyponyms, therefore not corrisponding.

Chapter V – Hypernymy

"All men are mortal. Socrates was mortal.

Therefore, all men are Socrates."

Woody Allen

This chapter describes the concept of hypernymy (Section 5.1) and its treatment in NLP, with a particular focus on distributional semantics (Section 5.2). After having provided such background, we introduce the most common metrics for hypernymy identification, presenting SLQS (Section 5.3), an unsupervised method for generality identification, which can be applied to identify hypernymy. SLQS was evaluated in two tasks on BLESS (Section 5.4), showing interesting results.

The chapter is an adaptation of:

- Santus, E., Lenci, A., Lu, Q., & Schulte im Walde, S. (2014a). Chasing Hypernyms in Vector Spaces with Entropy. Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014), 2, p. 38-42.

5.1 Hypernymy and Taxonymy

Hypernymy (sometimes also referred to as IS-A; Collins & Quillian (1969)), together with similarity, is the main organizer of the semantic memory (Murphy G. L., 2002): similarity is used to cluster together similar word meanings, while hypernymy organizes them hierarchically. In fact, as suggested by its name (i.e. the term *hypernym* comes from the Greek *hyper* = "over" *-onym* = "name", "higher-level name", and its counterpart *hyponym* comes from *hypo* = "under" *-onym* = "name", or "lower-level name"), hypernymy is the relation of dominance that structures semantic hierarchies (and, in some cases, taxonomies).

Hypernymy is not limited to structuring hierarchies. It is, for example, at the base of inferences, entailments, concept definitions and categorization process (Casagrande & Hale, 1967), which explains why it is learnt very early in childhood (Markman, 1981). For the same reason, it is "[by] far the most studied lexical relation in the computational community" (Pustejovsky, 1995). Four main properties of hypernymy have been identified in the literature (Cruse, 1986):

1. *Asymmetry*, or directionality: if *A* is a hypernym of *B*, *B* is not a hypernym of *A*, but it is its hyponym;
2. *Catenary*: if *A* is a hypernym of *B*, *B* can still be a hypernym of *C*, and so on;
3. *Transitivity*: if hypernymy exists between *A* and *B* and between *B* and *C*, hypernymy also exists between *A* and *C*.
4. *Inheritance Property*: the hyponym inherits the properties of the hypernym.

Taxonymy is argued to share the same properties of hypernymy, the only difference being that it requires keeping constant the relation of difference between the co-hyponyms (Cruse, 1986). For example, while a hierarchy can have one level based on type and one on gender differentiation (e.g. mammal → dog / human; and human → male / female), a taxonomy must keep the relation of difference constant in all levels (e.g. either type or gender). Cruse (2000) suggests that taxonymy is a prototypical type of hypernymy.

The concept of hypernymy can finally be sub-classified according to several factors. Miller (1998), for example, distinguished taxonomical (e.g. a *dog* IS-A *animal*) and functional hypernymy (e.g. *dog* IS-USED-AS-A *pet*). Herrmann & Herrmann (1984), on the other hand, based their distinction on the type of information involved: perceptual hyponymy (e.g. *animal/horse*), functional (e.g. *vehicle/car*), geographical (e.g. *country/China*), activity (e.g. *game/chess*), state (e.g. *emotion/fear*) and action (e.g. *cook/fry*).

5.2 Hypernymy and Taxonymy in NLP

The problem of identification of asymmetric relations like hypernymy and taxonymy has often been approached through semi-supervised methods, such as pattern-based (Hearst, 1992; Pennacchiotti & Pantel, 2006) and – more recently – machine learning (Weeds, Clarke, Reffin, Weir, & Keller, 2014; Levy, Remus, Biemann, & Dagan, 2015).

The identification of hypernymy and taxonymy has been addressed through unsupervised distributional approaches only in a limited way (Kotlerman, Dagan, Szpektor, & Zhitomirsky-Geffet, 2010), especially because it is not clear to what extent distributional similarity (which is by definition a symmetric relation) is appropriate to model the semantic properties of asymmetric relations. In fact, it is not enough to say that *animal* is distributionally similar to *dog*. We must also account for the fact that *animal* is semantically broader than *dog*: every *dog* is an *animal*, but not every *animal* is a *dog*.

The few work that has attempted at a completely unsupervised approach to the identification of hypernymy in corpora has mostly relied on some version of the *Distributional Inclusion Hypothesis* (DIH; Geffet & Dagan (2005); Weeds & Weir (2003); Weeds, Weir, & McCarthy (2004)), according to which the contexts of a narrower term are also shared by the broader term.

A measure formalizing the DIH is the *WeedsPrec* (Weeds & Weir, 2003; Weeds, Weir, & McCarthy, 2004)), which quantifies the weights of the features *f* of a narrower term *u* that are included into the set of features of a broader term *v*:

$$WeedsPrec(u,v) = \frac{\sum_{f \in F_u \cap F_v} w_u(f)}{\sum_{f \in F_u} w_u(f)} \qquad 13$$

where F_x is the set of features of a term *x*, and $w_x(f)$ is the weight of the feature *f* of the term *x*. This measure identifies the direction of hypernymy with 71% accuracy on word-pairs extracted from WordNet (Fellbaum C. , 1998). This result, however, was not significantly better than the frequency baseline, according to which more general words are more frequent.

Variations of this measure have been proposed. Clarke (2009) extended the DIH, suggesting that generality difference can be calculated as the degree to which the dimensions of the narrower term have lower values than the broader ones, across all the intersected dimensions. Lenci & Benotto (2012) adapted this measure to check not only to which extent the features of the narrower term are included in the features of the broader, and also how the features of the broader are not included in the features of the narrower. Kotlerman, Dagan, Szpektor, & Zhitomirsky-Geffet (2010) combined Average Precision (AP) with the balancing approach of Szpektor & Dagan (2008), outperforming the above mentioned methods. Herbelot & Ganesalingam (2013) measured the Kullback-Leibler (KL) divergence between the probability distribution over context words for a term, and the background probability distribution, based on the idea that hypernyms, being less informative words, should have smaller values for such divergence. Rimmel (2014) considered the top features in a context vector as topics and used a Topic Coherence (TC) measure.

In this chapter, we introduce *SLQS*, an entropy-based distributional measure that aims to identify hypernyms by providing a distributional characterization of their *semantic generality*. According to the *Distributional Informativeness Hypothesis* (DInH), the generality of a term can be inferred from the informativeness of its most typical linguistic contexts. We assess this hypothesis in two tasks: i) the identification of the broaderer term in hyponym-hypernym pairs (*directionality task*); ii) the discrimination between hypernymy and other semantic relations (*detection task*).

5.3 SLQS

DIH is grounded on an “extensional” definition of the asymmetric character of hypernymy: since the class denoted by a hyponym (i.e. extension) is included in the class denoted by the hypernym, hyponyms are expected to occur in a subset of the contexts of their hypernyms. However, it is also possible to provide an "intensional" definition of the same asymmetry. In fact, the typical characteristics making up the "intension" expressed by a hypernym (i.e. concept; e.g. *move* or *eat* for *animal*) are semantically more general than the characteristics forming the "intension" of its hyponyms (e.g. *bark* or *has fur* for *dog*). This corresponds to the idea that superordinate terms like *animal* are less informative than their hyponyms (Murphy G. L., 2002). From a distributional point of view, we can therefore expect that the most typical linguistic contexts of a hypernym are less informative than the most typical linguistic contexts of its hyponyms. In fact, contexts such as *bark* and *has fur* are likely to co-occur with a smaller number of words than *move* and *eat*.

Starting from this hypothesis, which we call *Distributional Informativeness Hypothesis* (DInH), and using entropy as an estimator of context informativeness (Shannon, 1948), we propose *SLQS*, which infers the semantic generality of a word from the median entropy of its statistically most prominent contexts.

For every term w_i we identify the N most associated contexts c (where N is a parameter empirically set to 50)[16]. The association strength has been calculated with *Local Mutual Information* (*LMI*; Evert (2005)). For each selected context c, we define its entropy $H(c)$ as:

[16] N=50 is the result of an optimization of the model against the dataset after trying the following suboptimal values: 5, 10, 25, 75 and 100.

$$H(c) = -\sum_{i=1}^{n} p(f_i|c) \cdot log_2\big(p(f_i|c)\big) \qquad 14$$

where $p(f_i|c)$ is the probability of the feature f_i given the context c, obtained through the ratio between the frequency of $<c, f_i>$ and the total frequency of c. The resulting values $H(c)$ are then normalized in the range 0-1 by using the Min-Max-Scaling: $H_n(c)$. Finally, for each term w_i we calculate the median entropy E_{wi} of its N contexts:

$$E_{w_i} = Me_{j=1}^{N}\left(H_n(c_j)\right) \qquad 15$$

E_{wi} can be considered as a *semantic generality index* for the term w_i: the higher the index value, the more semantically general w_i is. *SLQS* is then defined as the reciprocal difference between the semantic generality of two terms w_1 and w_2:

$$SLQS(w_1, w_2) = 1 - \frac{E_{w_1}}{E_{w_2}} \qquad 16$$

According to this formula, $SLQS<0$, if $E_{w1} > E_{w2}$; $SLQS\simeq 0$, if $E_{w1} \simeq E_{w2}$; and $SLQS>0$, if E_{w1} <. E_{w2}. *SLQS* is an asymmetric measure because, by definition, $SLQS(w_1,w_2) \neq SLQS(w_2,w_1)$ (except when w_1 and w_2 have exactly the same generality). Therefore, if $SLQS(w_1,w_2)>0$, w_1 is semantically less general than w_2.

5.4 Experiments

In the following subsections we describe the assessment of *SLQS* in two tasks (i.e. directionality identification and hypernymy identification), providing information about the training corpora, the DSM and the dataset.

5.4.1 Corpora, DSM and Dataset

For the experiments, we used a standard window-based DSM recording co-occurrences with the nearest 2 content words to the left and right of each target word. Co-occurrences were extracted from a combination of the freely available ukWaC and WaCkypedia corpora (see Section 2.3.1) and weighted with LMI.

To assess *SLQS* we relied on a subset of *BLESS* (Baroni & Lenci, 2011), a freely-available dataset that includes 200 distinct English concrete nouns as target concepts, equally divided between living and non-living entities (e.g. BIRD, FRUIT, etc.). For each target concept, *BLESS* contains several *relata*, connected to it through one relation, such as co-hyponymy (COORD), hypernymy (HYPER), meronymy (MERO) or no-relation (RANDOM-N)[17].

Since *BLESS* contains different numbers of pairs for every relation, we randomly extracted a subset of 1,277 pairs for each relation, where 1,277 is the maximum number of HYPER-related pairs for which vectors existed in our DSM.

[17] In these experiments, we only consider the *BLESS* pairs containing noun relata.

5.4.2 Task 1: Directionality Identification

In this experiment we aimed at identifying the hypernym in the 1,277 hypernymy-related pairs of our dataset. Since the HYPER-related pairs in *BLESS* are in the order hyponym-hypernym (e.g. *eagle-bird*, *eagle-animal*, etc.), the hypernym in a pair *(w_1,w_2)* is correctly identified by *SLQS*, if *SLQS (w_1,w_2)* > 0.

Following Weeds, Weir, & McCarthy (2004), we used word frequency as a baseline model. This baseline is grounded on the hypothesis that hypernyms are more frequent than hyponyms in corpora. Table 14 gives the evaluation results:

	SLQS	**WeedsPrec**	**BASELINE**
POSITIVE	1111	805	844
NEGATIVE	166	472	433
TOTAL	1277	1277	1277
PRECISION	87.00%	63.04%	66.09%

Table 14: Accuracy for Task 1.

As it can be seen in Table 14, *SLQS* scores a precision of 87% in identifying the second term of the test pairs as the hypernym. This result is particularly significant when compared to the one obtained by applying WeedsPrec (+23.96%). As it was also noticed by Geffet & Dagan (2005) with reference to a previous similar experiment performed on a different corpus (Weeds, Weir, & McCarthy (2004)), WeedsPrec in this task performs comparably to the frequency baseline. *SLQS* scores instead a +20.91% to it.

5.4.3 Task 2: Hypernymy Identification

The second experiment aimed at discriminating HYPER test pairs from those linked by other types of relations in *BLESS* (i.e., MERO, COORD and RANDOM-N). To this purpose, we assumed that hypernymy is characterized by two main properties: i) the hypernym and the hyponym are distributionally similar (in the sense of the *Distributional Hypothesis*), and ii) the hyponym is semantically less general than the hypernym. We measured the first property with the *vector cosine* and the second one with *SLQS*.

After calculating *SLQS* for all the pairs in our datasets, we set all the negative values to zero, that is to say those in which – according to *SLQS* – the first term is semantically more general than the second one. Then, we combined *SLQS* and *vector cosine* by taking their product. The greater the resulting value, the greater the likelihood that we are considering a hypernymy-related pair, in which the first word is a hyponym and the second word is a hypernym.

To evaluate the performance of *SLQS*, we used *Average Precision* (AP; see Section 4.4.2), a method derived from *Information Retrieval* that combines precision, relevance ranking and overall recall, returning a value that ranges from 0 to 1. AP=1 means that all the instances of a relation are at the top of the rank, whereas AP=0 means they are at the bottom. AP is computed for the four relations for which we extracted pairs from *BLESS*. *SLQS* was also compared with *WeedsPrec* and *vector cosine*, again using frequency as baseline. Table 15 shows the results.

	HYPER	**COORD**	**MERO**	**RANDOM**
Baseline	0.4	0.51	0.38	**0.17**
Cosine	0.48	0.46	**0.31**	0.21
WeedsPrec	0.5	0.35	0.39	0.21
SLQS * Cosine	**0.59**	**0.27**	0.35	0.24

Table 15: AP values for Task 2.

The AP values show the performances of the tested measures on the four relations. The optimal result would have been a score of 1 for HYPER and 0 for the other relations.

The product between *SLQS* and *vector cosine* gets the best performance in identifying HYPER (+0.09 in comparison to *WeedsPrec*) and in discriminating it from COORD (-0.08 than *WeedsPrec*). It also achieves better results in discriminating MERO (-0.04 than *WeedsPrec*). On the other hand, it seems to get a slightly lower precision in discriminating RANDOM-N (+0.03 in comparison to *WeedsPrec*). The likely reason is that unrelated pairs might also have a fairly high semantic generality difference, slightly affecting the measure's performance. Figure 6 gives a graphic depiction of the performances. *SLQS* corresponds to the black line in comparison to the *WeedsPrec* (black borders, grey fill), the *vector cosine* (grey borders) and the baseline (grey fill).

To conclude, our experiments demonstrate that hypernymy can be identified by measuring the generality spread between the words in the pairs with entropy. *SLQS* does not account however for the other property that characterizes hypernymy, that is similarity: it therefore needs to be combined with a similarity measure, such as vector cosine, when unrelated pairs need to be discriminated too.

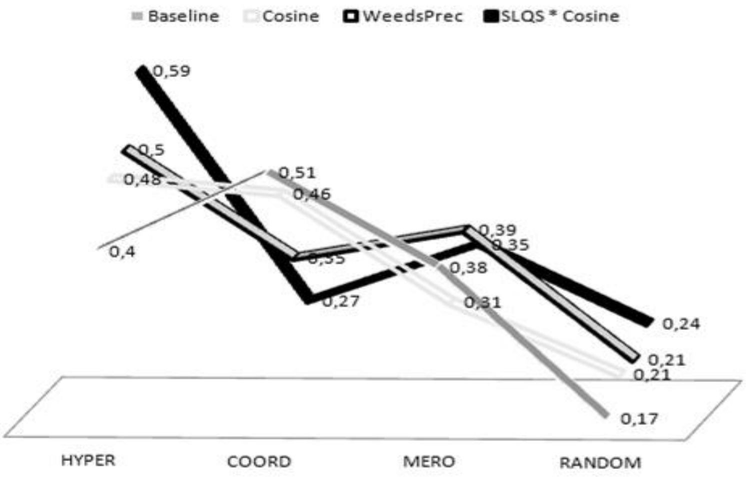

Figure 6: AP values for Task 2.

5.5 Summary of Chapter V

In this chapter, we have discussed the concept of hypernymy (Section 5.1) and described *SLQS* (Section 5.3), an asymmetric distributional measure of semantic generality which is able to identify the broader term in a hypernym-hyponym pair and, when combined with *vector cosine*, to discriminate hypernymy from other types of semantic relations. The successful performance of *SLQS* in the reported experiments confirms that hyponyms and hypernyms are distributionally similar, but hyponyms tend to occur in more informative contexts than hypernyms. *SLQS* shows that an “intensional” characterization of hypernymy can be pursued in distributional terms. This opens up new possibilities for the study of semantic relations in DSMs.

Chapter VI – Supervised Learning of Hierarchies

"*Knowledge is knowing a tomato is a fruit;*

wisdom is not putting it in a fruit salad."

Anonymous

In this chapter, we describe *ROOT9* (Section 6.3), a supervised method based on a *Random Forest* algorithm and nine corpus-based features (mostly inspired at the previously described unsupervised measures: see Chapters III, IV and V) for the identification of hierarchical relations (Section 6.1), namely hypernymy and co-hyponymy, as opposites of unrelated words. The method is evaluated in three tasks (Section 6.4) and shows competitive performance with the state-of-the-art.

The chapter is an adaptation of:

- Santus, E., Lenci, A., Chiu, T.-S., Lu, Q., & Huang, C.-R. (2016e). Nine Features in a Random Forest to Learn Taxonomical Semantic Relations. Portorož, Slovenia: Proceedings of Language Resources and Evaluation Conference (LREC 2016).
- Santus, E., Lenci, A., Chiu, T.-S., Lu, Q., & Huang, C.-R. (2016f). ROOT13: Spotting Hypernyms, Co-Hyponyms and Randoms. Phoenix, Arizona: Proceedings of Association for the Advancement of Artificial Intelligence (AAAI).

6.1 Hierarchical Discrimination

Distinguishing hypernyms from co-hyponyms and, thus, discriminating them from semantically unrelated words (henceforth *randoms*) is a fundamental task in *Natural Language Processing* (NLP). As we have seen in 5.1, hypernymy represents a key organization principle of semantic memory (Murphy G. L., 2002), the backbone of taxonomies and ontologies, and one of the crucial semantic relations supporting lexical entailment (Geffet & Dagan, 2005). Co-hyponymy (or *coordination*), on the other hand, is the relation held by words sharing a close hypernym, which are therefore attributionally similar (Weeds, Clarke, Reffin, Weir, & Keller, 2014).

The ability of discriminating hypernymy, co-hyponymy and random words has numerous applications, including *Automatic Thesauri Creation*, *Paraphrasing*, *Textual Entailment*, *Sentiment Analysis* and so on (Weeds, Clarke, Reffin, Weir, & Keller, 2014; Xu, Santus, Laszlo, & Huang, 2015). For this reason, in the last decades, numerous methods, datasets and shared tasks have been proposed to improve computer ability to discriminate such relations, generally achieving promising results (Santus, Lenci, Chiu, Lu, & Huang, 2016e-f; Roller, Erk, & Boleda, 2014; Weeds, Clarke, Reffin, Weir, & Keller, 2014; Santus, Lenci, Lu, & Schulte im Walde, 2014; Levy, Remus, Biemann, & Dagan, 2015; Geffet & Dagan, 2005; Lenci & Benotto, 2012; Weeds, Weir, & McCarthy, 2004; Rimmel, 2014). Both supervised and unsupervised approaches have been investigated. The former have been shown to outperform the latter in Weeds, Clarke, Reffin, Weir, & Keller (2014), even though Levy, Remus, Biemann, & Dagan (2015) have claimed that – because of lexical memorization – these methods may learn whether a term *y* is a prototypical hypernym, regardless of its actual relation with *x*.

In this chapter we report a revision of *ROOT13* (Santus, Lenci, Chiu, Lu, & Huang, 2016f), a supervised method based on a *Random Forest* algorithm (Breiman, 2001) and thirteen corpus-based features. The feature contribution is evaluated with an ablation test, using a 10-fold cross validation on 9,600 pairs[18] randomly extracted from *EVALution* (Santus, Yung, Lenci, & Huang, 2015b), *Lenci/Benotto* (Benotto, 2015) and *BLESS* (Baroni & Lenci, 2011). The ablation test has shown that four out of thirteen features were actually not contributing to the system's performance, and they were therefore removed, turning *ROOT13* into *ROOT9*. On the 9,600 pairs, *ROOT9* achieved an *F1 score* of 90.7% when the three classes were present, 95.7% when we had to discriminate hypernyms and co-hyponyms, 91.8% for hypernyms and randoms, and 97.8% for co-hyponyms and randoms.

In order to compare *ROOT9* with the state-of-the-art, we have also evaluated it in the Weeds, Clarke, Reffin, Weir, & Keller (2014)'s datasets. Unfortunately, *ROOT9* was not able to cover the full datasets, as several words in their pairs were missing from our *Distributional Semantic Model* (DSM) because of their low frequency. Nevertheless, the authors kindly provided the results of their models on our subsets, so that the comparison can be considered reliable. Also in relation to the state-of-the-art, *ROOT9* is proved to be competitive, being slightly outperformed in all the datasets only by the *svmCAT* model (Weeds, Clarke, Reffin, Weir, & Keller,

[18] The 9,600 pairs are available at https://github.com/esantus/ROOT9

2014), which is a *Support Vector Machine* (SVM) classifier run on the concatenation of the dependency-based distributional vectors of the words in the pairs.

Finally, we carried out an extra test to verify whether the system was actually learning the semantic relation between two word pairs, or simply identifying prototypical hypernyms (Levy, Remus, Biemann, & Dagan, 2015). The test consisted in providing to the trained model switched hypernyms (e.g. from "*dog HYPER animal*" to "*dog RANDOM fruit*"), and verify how they were classified. Our results show that most of the switched hypernyms were in fact misclassified as hypernyms (especially when those hypernyms were in the training test), and that the only way to overcome such problem is to explicitly provide the model with negative examples (i.e., switched hypernyms tagged as randoms) during the training.

6.2 Related Work

Since the pioneering work of Hearst (1992), who used a pattern based approach for the "automatic acquisition of hyponyms from large text corpora", a large number of distributional methods have been applied to the identification of hypernyms.

Among the supervised methods, Baroni, Bernardi, Do, & Shan (2012) proposed to use an SVM classifier on the concatenation (after having tried also subtraction and division) of the vectors. Roller, Erk, & Boleda (2014) used the vectors' difference, while Weeds, Clarke, Reffin, Weir, & Keller (2014) implemented numerous combinations (difference, multiplication, sum, concatenation, etc.), comparing them against the most common unsupervised methods. The authors demonstrated that supervised methods generally perform better than unsupervised ones, but they acknowledge that these methods tend to learn ontological information, re-using it any time a word occur again in the dataset. For this reason, they suggest to adopt a new dataset, where words occur at most twice, once per side. Weeds, Clarke, Reffin, Weir, & Keller (2014)'s observation was further investigated by Levy, Remus, Biemann, & Dagan (2015), who claimed that supervised methods learn whether a term y is a prototypical hypernym, regardless of its actual relation with x.

6.3 ROOT9

ROOT13 was firstly introduced in Santus, Lenci, Chiu, Lu, & Huang (2016f). It uses the *Random Forest* algorithm implemented in *Weka* (Witten, Frank, & Hall, 2005), with the default settings (i.e., 100 trees, 1 seed, and *maxDepth* and *numFeatures* initialized to 0), and relies on thirteen features that are carefully described below. Each of them is automatically extracted from a window-based DSM, trained on a combination of ukWaC and WaCkypedia corpora (see Section 2.3.1), counting word co-occurrences within the 5 nearest content words to the left and right of each target. Only adjectives, nouns and verbs with frequency above 1,000 are included in the DSM. As it will be shown in the evaluation, four out of thirteen features were redundant and were not contributing to the system performance. They were therefore dropped, turning *ROOT13* into *ROOT9*.

6.3.1 Features

The feature set was designed to identify several distributional properties characterizing the terms in the pairs. On top of the standard distributional features (e.g., *co-occurrence frequency* and *words frequencies*), we have added several types of information that have been proved to be effective to discriminate paradigmatic semantic relations in vector spaces (see chapters III, IV and V). All the features were computed using the above-mentioned DSM and normalized in the range 0-1.

6.3.1.1 Co-Occurrence

Cooc is defined as the co-occurrence frequency between the two terms in the pair, within the DSM window. According to the *Co-occurrence Hypothesis* (Miller & Charles, 1991), this measure is discriminative for synonyms and antonyms: antonyms are in fact expected to occur in the same sentence more often than synonyms. Since co-hyponyms can often be seen as a specific kind of opposition (e.g. “Winter or summer?”; Murphy (2003)), this measure should help in discriminating them from hypernyms and randoms (Santus, Lu, Lenci, & Huang, 2014a).

6.3.1.2 Frequency

Frequency is an important property of words. Weeds & Weir (2003), for example, have shown that the frequency baseline was very competitive in identifying the directionality of hypernymy-related pairs. We can therefore expect that hypernyms have higher frequency than hyponyms. Frequency is incorporated in our model with three features, namely one for each word involved in the pair (*Freq1,2*), plus one used for storing the difference between the frequencies (*Diff Freq*).

6.3.1.3 Entropy

As we have seen in 5.3, *entropy* is generally used to measure informativeness: the lower the entropy of an event, the higher its informativeness (see Section 5.3). Words in a corpus has very low entropy, as every word needs to fulfil specific morphological, syntactic and semantic requirements in order to occur in specific positions (e.g. in a phrase like “x barks”, it is very likely that x is “dog”, because x is expected to be a noun, and only dogs are known for barking). Nevertheless, word entropies vary according to several factors, such as the generality and prototypicality of the word. As claimed by Murphy (2002), the amount of informativeness in the taxonomies increases, when moving from the superordinate to the subordinate level. We can therefore use entropy as an index of word informativeness. It is calculated using the Shannon (1948)’s equation, as described in the Equation 14 in Section 5.3, reported here for simplicity:

$$H(w) = -\sum_{i=1}^{n} p(c_i|w) \cdot log_2\big(p(c_i|w)\big) \qquad 14$$

where $p(c_i|w)$ is the probability of the context c_i given the word w, computed as the ratio between the co-occurrence frequency of the pair $<w, c_i>$ and the total frequency of w.

In our system, entropy corresponds to three features, namely one for each word in the pair (*Entr1,2*), plus one used for storing the difference between the entropies (*Diff Entr*).

6.3.1.4 Shared and APSyn

Shared and *APSyn* (see 3.2.1) are two features that do not rely on the full distribution of the words, but on the top N most related contexts to the words in a pair, where N was empirically fixed at 1000 (see sections 2.3.3.2 and 3.1.1 for the motivations behind feature selection).

We calculated *APSyn* using *Positive Pointwise Mutual Information* (PPMI; Levy, Goldberg, & Dagan (2015)), as it has shown some improvements. Once the PPMI values are assigned to all contexts of the target words (i.e. the words in the pair), we rank these contexts in a decreasing order, and consider only the top N, with $N = 1000$. At this point, *Shared* is the cardinality of the intersection of the top N contexts of the target words. *APSyn*, instead, is defined like in Section 3.2.1, namely the weighted cardinality of the intersection, where the weight is the average ranking of the common features, as in Equation 11, reproduced below:

$$APSyn(w_1, w_2) = \sum_{f \in N(F_1) \cap N(F_2)} \frac{1}{(rank_1(f) + rank_2(f))/2} \qquad 11$$

That is, for every feature f included in the intersection between the top N features of w_1, $N(F_1)$, and w_2, $N(F_2)$, *APSyn* will add 1 divided by the average rank of the feature, among the top PPMI ranked features of w_1, $rank_1(f_1)$, and w_2, $rank_2(f_2)$.

6.3.1.5 Contexts Frequency

We have noticed that hypernyms tend to occur in more frequent contexts than co-hyponyms and randoms. Our system exploits two features, *C-Freq1,2*, capturing the average frequency of the *N* top contexts of the target words in the pair.

6.3.1.6 Contexts Entropy

Given what mentioned in Section 5.2, the DIH and the DInH (Weeds & Weir, 2003; Santus, Lenci, Lu, & Schulte im Walde, 2014a), general words are likely to occur in a larger variety of contexts (i.e. higher frequency) and in more general ones (i.e. less informative), compared to specific words. In fact, although hypernyms can certainly occur in more selective contexts, specific words are more likely to be chosen in these situations. Consider the following sentences:

a) The *X* has barked all night.

b) The *Y* has arrested the thieves.

Any reader would agree that *X* is likely to be *dog* and *Y policeman*. Of course, *X* could have also been *animal* and *Y man*, or – even – both *X* and *Y* could have been *mammal*, but we expect that such general words are less frequently used in these contexts, as their hyponyms are more appropriate.

Adopting a similar approach to the one described in Seciton 5.3, we have measured the average entropy of the top *N* most related contexts and used it as an index of generality. Entropy is again calculated as in Equation 14 (see Section 5.3) and the most related contexts are identified by sorting the context list by PPMI. The higher the average entropy of the top-*N* contexts, the less informative the word (i.e. it is more likely to be a hypernym). Our system uses one of these features for each target: *C-Entr1,2*.

6.4 Experiments

We have performed three tasks: i) an ablation test to evaluate the contribution of the features on our dataset (henceforth, *ROOT9 Dataset*; see Section 6.4.4); ii) an evaluation against the state-of-the-art, and – in particular – against the best performant models in Weeds, Clarke, Reffin, Weir, & Keller (2014) (see Section 6.4.5); iii) an evaluation on switched pairs to verify whether it was learning the actual semantic relations or the prototypical hypernyms (Levy, Remus, Biemann, & Dagan, 2015) (see Section 6.4.6).

We performed the ablation test on a tree-classes classification task (hypernyms, co-hyponyms and randoms), removing one feature at a time and measuring the loss/gain (*F1 score* is used for the evaluation on a 10-fold cross validation). By doing so, we have found that four of our features were in fact redundant, and we have therefore removed them from the final model (hence the name *ROOT9* as opposed to *ROOT13*). Once the best model has been identified, we have performed three binary classification tasks, involving only two classes per time. *F1 score* on a 10-fold cross validation was chosen as accuracy measure.

The second task was a binary classification tasks on the four datasets proposed by (Weeds, Clarke, Reffin, Weir, & Keller, 2014). These datasets are described below, in Section 6.4.2. The task allowed us to compare *ROOT9* against the state-of-the-art models reported by Weeds and colleagues.

The last task was performed on an extended *ROOT9 Dataset*, including also 3,200 randomly switched hypernyms to verify whether they were classified as hypernyms or as randoms.

6.4.1 ROOT9 Dataset

We have used 9,600 pairs, randomly extracted from three datasets – *EVALution* (Santus, Yung, Lenci, & Huang, 2015b), *Lenci/Benotto* (Benotto, 2015) and *BLESS* (Baroni & Lenci, 2011) –, which are freely available at https://github.com/esantus/ROOT9. The pairs are equally distributed among the three classes (i.e., hypernyms, co-hyponyms and random words) and involve several Parts-Of-Speech (i.e., adjectives, nouns and verbs).

The class of hypernyms contains 2,447 noun pairs, 458 verb pairs and 295 adjective pairs. The class of co-hyponyms has only 3,200 noun pairs, which were completely derived from BLESS, as this relation does not exist in the other two datasets. The class of randoms contains 1,100 noun pairs, 1,050 verb pairs and 1,050 random pairs.

The full dataset contains 4,263 terms (2,380 nouns, 958 verbs and 927 adjectives), so that every term occurs on average 4.5 times. Considering only the first word in the pairs, we have 1,265 different terms (987 nouns, 186 verbs and 92 adjectives). Considering instead only the second word, we have 3,665 terms (1,945 nouns, 860 verbs and 862 adjectives).

In the third task, we have extended this dataset randomly switching the 3,200 hypernymy pairs (e.g. from "*car HYPER vehicle*" to "*car RANDOM mammal*") to verify whether ROOT9 was able to classify them as randoms.

6.4.2 Weeds Datasets

In order to compare *ROOT9* to the state-of-the-art, we have evaluated it with the datasets created by Weeds, Clarke, Reffin, Weir, & Keller (2014).[19] These are four datasets, containing respectively: i) hypernyms versus other relations (extracted from WordNet; henceforth *WN Hyper*); ii) co-hyponyms versus other relations (extracted from WordNet; henceforth *WN Co-Hyp*); iii) hypernyms versus other relations (extracted from BLESS; henceforth *Bless Hyper*); iv) co-hyponyms versus other relations (extracted from BLESS; henceforth *Bless Co-Hyp*).

The *WN* dataset – which includes both *WN Hyper* and *WN Co-Hyp* – in particular, was built after noticing that supervised systems tended to perform well also when running on random vectors. This happens because they are able to learn ontological information and re-use it whenever the words re-appear in other pairs. For this reason, the authors have constructed a dataset where words occurred at most twice (once on the left and once on the right of the relation). In this dataset,

[19] The datasets are freely available at: https://github.com/SussexCompSem/learninghypernyms

ontological information cannot be learnt and re-used, and indeed the random vectors cannot perform well.

Unfortunately our DSM did not cover the whole datasets, because of the chosen frequency threshold (in Table 16, we report the size of our subsets in comparison to the original datasets). However, Weeds and colleagues kindly provided the results of their models on our subsets, so that the comparison is representative[20].

	WN Hyper	**WN Co-Hyp**	**Bless Hyper**	**Bless Co-Hyp**
Weeds et al.	2514	4166	1668	5835
Subset	1791	2936	1636	5389
Coverage %	71.24	70.47	98.08	92.36

Table 16: Coverage on Weeds et al. (2014)'s datasets.

6.4.3 Baselines and Other Models

For our internal tests (Task 1, Section 6.4.4), we have implemented two baselines, which can be used as reference for evaluating the performance of *ROOT9*: *COSINE* and *RANDOM13*.

The first baseline simply uses the *vector cosine* (*COSINE*) as feature of a *Random Forest* classifier in the default settings (i.e. 100 trees, 1 seed, and *maxDepth* and *numFeatures* initialized to 0). This baseline is supposed to perform particularly

[20] The subsets of Weeds, Clarke, Reffin, Weir, & Keller (2014)'s datasets are also available at https://github.com/esantus/ROOT9.

well in discriminating similar words (i.e. hypernyms and co-hyponyms) from randoms.

The second baseline (*RANDOM13*) relies on a default *Random Forest* classifier, but uses thirteen randomly initialized features, with values between 0 and 1.

While the vector cosine achieves a reasonable accuracy, which is anyway far below the results obtained by our model, the random baseline performs much worse. The discrepancy with what was found by Weeds, Clarke, Reffin, Weir, & Keller (2014) – namely that random vectors perform particularly well when words are re-used in the dataset – may depend on the small number of features, which does not allow the system to identify discriminative random dimensions.

In the second task, we have used as baselines the most competitive models reported in Weeds, Clarke, Reffin, Weir, & Keller (2014), namely the SVM classifiers trained on the PPMI vector of the second word (*svmSINGLE*), or on the concatenated (*svmCAT*), summed (*svmADD*), multiplied (*svmMULT*) and subtracted (*svmDIFF*) PPMI vectors of the words in the pair. Such vectors contain as features all major grammatical dependency relations involving open class parts-of-speech. As a reference, we also report the performance of three main unsupervised methods: cosine, balAPinc (Kotlerman, Dagan, Szpektor, & Zhitomirsky-Geffet, 2010) and invCL (Lenci & Benotto, 2012). A threshold *p*, empirically found in a training set, was used in these methods for the decision.

6.4.4 Task 1: Ablative and Binary Tests

Table 17 describes the feature contribution in the ablation test. Given the set of thirteen features of ROOT13 (Santus, Lenci, Chiu, Lu, & Huang, 2016f), we have removed each of them and measured the loss (negative) or the gain (positive).

As shown in Table 16, most of the features are contributing for an increment between 1.12% and 2.46%. The highest contribution comes from the *C-Entr1,2*, which were inspired by *SLQS* (see Chapter V), and the second highest contribute is given by *APSyn* (see Chapter III). Interestingly, four of the thirteen features were not contributing to the performance, thus penalizing our system between 0.11% and 0.34%. These features are *Diff Entr*, *Diff Freq*, *Co-Occurrence*, and *APSyn* and *Shared*, when they are used together (so we kept only *APSyn*, removing *Shared*). The main reason why these features negatively affect the results could be due to the fact that they contain redundant information. If we remove both *APSyn* and *Shared*, for example, we have a loss of 1.79%, but when we remove only one of them we have a gain of 0.34%. In a similar way, *Diff Entr* and *Diff Freq* can be seen as redundant in respect to the features *Entr1,2* and *Freq1,2*. Perhaps surprisingly, *Cooc* does not contribute to the final score, but instead lowers it.

	Hyper Co-Hyp Random	**LOSS OR GAIN**
ROOT13	**89.3**	**0.00%**
- C-Freq 1, 2	88.2	-1.23%
- C-Entr 1, 2	87.1	-2.46%
- APSyn	89.6	0.34%
- Shared	89.6	0.34%
- Shared + APSyn	87.7	-1.79%
- Diff Entr	89.6	0.34%
- Diff Freq	89.7	0.45%
- Entr 1, 2	88.0	-1.46%
- Freq 1, 2	88.3	-1.12%
- Cooc	89.4	0.11%
ROOT9	**90.7**	**1.12%**
BASELINES		
ROOT9 using SMO	68.6	-23.18%
ROOT9 using Logistic	73	-18.25%
COSINE	57.2	-35.95%
RANDOM13	33.4	-62.60%

Table 17: Ablation test, *F1 scores* on a 10-fold cross validation and loss/gain values. Scores are in percent.

Removing the four redundant features (we removed *Shared* but we kept *APSyn*), *ROOT13* turns into *ROOT9*. This system outperforms all the baselines (i.e. *COSINE*, *RANDOM13*) and *ROOT13*. For the sake of completeness, in Table 18 we also report the performance of ROOT9 using *Logistic Regression* and *SMO* classifiers. As it can be seen, the *Random Forest* version largely outperforms the other classifiers in this dataset. However, it is worth noticing here that such difference disappears with the *WN* datasets proposed by Weeds and colleagues (see Table 19).

	Hyper	**Hyper**	**Co-Hyp**
	Co-Hyp	**Random**	**Random**
ROOT13	94.3	91.1	97.4
ROOT9	**95.7**	**91.8**	**97.8**
- using SMO	77.3	80.1	93
- using Logistic	78.7	82.1	95.3
COSINE	69.8	64.1	79.4
RANDOM13	50.1	49.6	51.4

Table 18: *F1 scores* on a 10-fold cross validation for binary classification tasks. Scores are in percent.

Table 18 describes the results of *ROOT9* and the baseline in the binary classification tasks. These results confirm the analysis suggested above.

6.4.5 Task 2: ROOT9 vs. State-of-the-art

In Table 19, we show *ROOT9* performance compared to the best systems reported by Weeds, Clarke, Reffin, Weir, & Keller (2014). The scores are all calculated on subsets of Weeds and colleagues' datasets, as reported in Section 4.3.

Considering all the datasets, *ROOT9* is the second best performing system, after *svmCAT* (Weeds, Clarke, Reffin, Weir, & Keller, 2014), which uses the SVM classifier on the concatenation of PPMI vectors, containing as features all major grammatical dependency relations involving open class parts-of-speech.

The SVM classifier on the sum (*svmADD*) and the multiplication (*svmMULT*) of the same PPMI vectors performs better in identifying co-hyponyms, but worst in identifying hypernyms. The SVM on the difference (*svmDIFF*) and on the second PPMI vector (*svmSINGLE*) is instead particularly good at identifying hypernyms, while it performs poorly at identifying co-hyponyms.

Among the unsupervised methods, we report the results for the *cosine* and the methods of Lenci and Benotto (2012; *invCL*) and Kotlerman, Dagan, Szpektor, & Zhitomirsky-Geffet, (2010; *balAPinc*). Such methods classify the pairs using the best threshold *p* observed in the training sets. In general, unsupervised methods are less competitive.

Differently from what observed in Section 6.4.3, the performance of *ROOT9* does not change by adopting a different classifier (i.e., *Random Forest*, *SMO* or *Logistic Regression*) on the *WN Hyper* and *WN Co-Hyp* datasets. However, it drastically changes again on the *BLESS Hyper* and *BLESS Co-Hyp* datasets. This may depend on the ability of the *Random Forest* classifier to learn more ontological information than *SMO* and *Logistic Regression*, even when the number of features is small.

	WN Hyper	**WN Co-Hyp**	**Bless Hyper**	**Bless Co-Hyp**
ROOT9	69.8	60.8	94.6	87.7
- using SMO	67.7	60.9	65.5	70.4
- using Logistic	68.8	61.2	65.5	71.9
STATE-OF-THE-ART (Weeds, Clarke, Reffin, Weir, & Keller, 2014)				
svmCAT	74.1	62.9	96.7	90.7
svmADD	40.9	66	68.5	94.1
svmMULT	40.3	63.2	75.1	96.4
svmDIFF	74.1	40.7	86.5	56.7
svmSINGLE	66.3	58.2	97.8	62.8
cosine	58.7	52.8	64.7	78.5
balAPinc	55.8	53.4	65.7	76.8
invCL	60.7	61.7	72.5	63.2

Table 19: *F1 scores*, in percent, on a 10-fold cross validation (state-of-the-art models are evaluated on a 5-fold cross validation).

6.4.6 Task 3: Learning Prototypical Hypernyms?

Finally, we have tried to test Levy, Remus, Biemann, & Dagan (2015)'s claim by evaluating the classifier on a dataset containing 3,200 hypernyms and 3,200 switched hypernyms (e.g. *apple* RANDOM *animal* and *dog* RANDOM *fruit*). In this evaluation, we have noticed that a large number of the switched hypernyms were indeed misclassified as hypernyms (up to 100% of them, if the words in the testing switched pairs were exactly the same as the ones used as hypernyms in the training set). In the attempt of correcting the behavior of the classifier, we extended the original 9,600 pairs dataset with other 3,200 switched hypernyms pairs labeled as randoms. It is important to notice that the switched hypernyms (tagged as *randoms*) contain the same words used in for the real hypernyms, and that in this new dataset, the size of the random class is double the others, including a total of 6,400 pairs. The new 10-fold cross validation test on the three classes registered a significant loss, passing from 90.7% to 84%. However, only 576 out of 6,400 randoms (most of which are likely to be the switched pairs) were misclassified as hypernyms.

These results confirm the lexical memorization and support the idea that future systems for the identification of semantic relations might benefit from relying on features that better represent the relations (e.g. lexical-syntactic patterns) rather than only the lexical properties of the words in the pairs.

6.5 Summary of Chapter VI

In this chapter, we have described *ROOT9* (Section 6.3), a classifier for hypernyms, co-hyponyms and random words that is derived from an optimization of *ROOT13* (Santus, Lenci, Chiu, Lu, & Huang, 2016f). The classifier, based on the Random Forest algorithm, uses only nine unsupervised corpus-based features, which have been described (Section 6.3.1), and whose contribution has been assessed (Section 6.4.4). The impressive results in our dataset (Section 6.4.1 and 6.4.4), developed by randomly extracting 9,600 pairs from *EVALution* (Santus, Yung, Lenci, & Huang, 2015), *Lenci/Benotto* (Benotto, 2015) and *BLESS* (Baroni & Lenci, 2011), were further tested against the state-of-the-art models presented in Weeds, Clarke, Reffin, Weir, & Keller (2014). The comparison has shown that *ROOT9* is in fact competitive with the state-of-the-art, being outperformed on all the datasets only by an SVM trained on concatenated dependency-based PPMI vectors. Interestingly, while on our dataset and on BLESS the chosen classifier is fundamental for the performance, on the WN Hyper and WN Co-Hyp datasets, Random Forest, SMO and Logistic Regression algorithm achieved a similar performance, suggesting that Random Forest performs better when can relies on lexical memorization, but in a comparable way to other classification methods otherwise.

Finally, we have noticed the effect reported in Levy, Remus, Biemann, & Dagan (2015). However, we managed to reduce it by training the model also on negative examples, namely switched hypernyms labeled as randoms (e.g. *apple* RANDOM *animal*, *dog* RANDOM *fruit*). These results confirm the lexical memorization and support the idea that future systems for the identification of semantic relations might benefit from relying on features that better represent the relations (e.g. lexical-syntactic patterns) rather than only the lexical properties of the words in the pairs.

With respect to the state-of-the-art, *ROOT9* shows that a few carefully designed features may reach the same discrimination power as thousands of distributional features. This suggests that combining them might lead to even better results. On top of it, *ROOT9* might be useful when memory is very limited. Finally, we would like to point out that all our features were extracted from a window based DSM (see Chapter II). It is possible that extracting the same features from a dependency-based DSM could have led to better results.

Conclusions

In the previous chapters we have discussed the importance of models of semantic memory (Introduction and Chapter I), and we have claimed that semantic relations are a fundamental building block for such models (Murphy M. L., 2003). For this reason, we have developed and evaluated several new methods for their automatic identification in corpora, namely three unsupervised methods (APSyn for similarity: see Chapter III; APAnt for opposition: see Chapter IV; and SLQS for hypernymy: see Chapter V) and a supervised one (ROOT9 for classifying hypernyms, co-hyponyms and randoms: see Chapter VI).

Our approaches rely on the framework of distributional semantics (Miller & Charles, 1991), which was carefully described in its cognitive and computational aspects in Chapter II. In particular, we have shown how such framework has been exploited in NLP to develop efficient representations of word meanings, starting from word co-occurrences in corpora (Harris, 1954).

Each method was introduced in a specific chapter, which provided the background about the relevant relation (i.e. similarity, opposition, hypernymy and taxonomical relations) and about its treatment in NLP. Each chapter then described the hypothesis, the method implementation and its evaluation. Our methods were shown to be competitive or even to outperform several state-of-the-art models. This has confirmed the validity of our hypotheses.

Since their performance and their contribution was analysed and discussed in their respective chapters, here we only summarize the major conclusions we can draw from our results:

1. Similarity: similar words not only occur in similar contexts, but they also share a larger amount of their most relevant contexts, compared to simply associated words (see Chapter III). This hypothesis was proved to be valid by showing that *APSyn* outperforms *vector cosine* in almost all settings, except when the latter is used on a PPMI-SVD reduced matrix, which is known to be the best setting in the literature. This is certainly due however to the SVD reduction rather than to the good performance of *vector cosine*. A consequence is that future research should focus on the extension of *APSyn* or of its principles, so that such rank-based measure can be also applied to a SVD reduced matrix. In order to understand whether vector cosine calculated on the top-N contexts would have performed better, in 3.3.6 we have developed two new versions of *vector cosine*, respectively calculated on i) the intersected contexts – as for *APSyn*; and ii) the unified ones. In both cases, *APSyn* was still the best performant measure, demonstrating that not only the feature selection but also the rank-based calculation contribute to the results, and suggesting therefore that *APSyn* and *vector cosine* might capture different aspects of similarity and that their future combination might lead to better results.

2. Opposition: not differently from synonyms, opposites share many contexts, but – compared to them – they share a smaller amount of relevant contexts (see Chapter IV). This claim was tested by evaluating the discriminative power of *APAnt* (i.e. the inverse of *APSyn*), which is likely to assign high scores to pairs that share few relevant contexts and lower scores to pairs that share many relevant contexts (notice that such measure needs to be calculated on related pairs – e.g. pairs with high *vector cosine* –, as unrelated words are expected to share very few contexts). Our tests demonstrated that APSyn is in fact more discriminative than i) a baseline implementing the co-occurrence hypothesis (i.e. the hypothesis that antonyms co-occur in the same sentence more often than by chance) and i) *vector cosine*. *APAnt* is not anyhow similar to the inverse of *vector cosine*, as it only verifies that the most relevant contexts are not shared, while the inverse of *vector cosine* would rather check that the vectors have very little correlation. This claim, which is based on a logical reasoning, should be proved in further studies.
3. Hypernymy: hypernyms are semantically more general than hyponyms; the generality of these words can be estimated in terms of informativeness, by measuring the entropy of their most relevant contexts. The higher the entropy of the most relevant contexts, the lower the informativeness of a given word, and – therefore – the higher its generality (see Chapter V). This hypothesis was proved by testing *SLQS*,

an entropy based measure that – in fact – compares the median entropy of the most relevant contexts of the two target words, in several tasks: directionality detection and hypernymy discrimination. *SLQS* performs better than other measures developed for the same purpose (e.g. WeedsPrec) on our dataset (BLESS), showing that there is in fact a generality spread between the words and that such spread can be measured in terms of informativeness. *SLQS* does not capture, however, similarity. Therefore it is expected to be combined with vector cosine or run only on related word pairs (e.g. pairs that are expected to have a high *vector cosine*).

4. Taxonomical relations: the distributional properties that were identified by our unsupervised measures can be used to train a supervised model (i.e. a *Random Forest algorithm*) to discriminate taxonomical semantic relations (i.e. hypernymy and co-hyponymy). Interestingly, nine unsupervised features are sufficient to achieve competitive performance with the thousands of features used by the state-of-the-art systems. In both cases, however, supervised methods seem to learn word properties (i.e. some regions of the vectors) rather than relation properties. In the future, therefore, they might benefit from features that are more oriented to represent relation properties rather than the lexical properties, such as lexical-syntactic patterns.

On top of these findings, it is worth mentioning two main characteristics of the proposed unsupervised methods. First, they have shown that it is possible to work with a subset of contexts (i.e. the most salient ones), which are assumed to be more informative than the full distribution of contexts. Furthermore, such selection reduces the computational load and complies with a principle of cognitive economy. Second, our methods are strongly grounded on cognitive and linguistic observations (Cruse, 1986; Murphy G. L., 2002). The findings can therefore be also considered from a theoretical point of view, and not only from the applicative perspective, and they can eventually contribute to the relative cognitive and linguistic theories, providing more background for further research.

In the next studies, we would like to target three major limitations of this thesis:

i) the systematic study of the hyperparameters (e.g. the impact of *N* and of different context types, such as dependency-based and joint-based); ii) the merging of the methods for developing a multi-class classification algorithm; and iii) the adaptation of the methods (and/or their principles) to reduced matrices (see Turney & Pantel (2010)) and *word embeddings* (Mikolov, Yih, & Geoffrey (2013)).

By i), we would like to test the measures on a larger amount of DSMs, evaluating all their hyperparameters and assessing different context types (e.g. dependency based and joint based), as well as providing a deeper error analysis, so that the methods can be refined and improvements can be adopted. Point ii) would instead target the very ambitious development of a multi-class classifier. So far, very few systems for multi-class classification were developed, as the task is particularly hard to tackle. A recent shared task has tried to promote this kind of approach to semantic relations (i.e. synonyms, antonyms, hypernyms and part-whole meronyms), and the best supervised system obtained 44.5% F1 score (Santus, Gladkova, Evert, &

Lenci, 2016), demonstrating that a lot of work is still needed in this direction. For what concerns point iii), given the recent introduction of models based on word embeddings and given the best performance of *vector cosine* on SVD reduced matrices, it becomes fundamental that our methods can be applied to such resources. At the moment, however, it is not yet clear whether this is possible, as our methods were designed for linguistic contexts and the features in such resources do not really represent linguistic contexts, being furthermore hard to interpret. This possibility might lead to even better results, allowing therefore the adoption of our methods (or their principles) in a larger range of applications.

Bibliography

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M., & Soroa, A. (2009). A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. Proceedings of North American Association for Computational Linguistics (NAACL-HLT, 2009).

Aston, G., & Burnard, L. (1998). *The BNC handbook: exploring the British National Corpus with SARA*. Edinburgh, Scotland: Edinburgh University Press.

Baroni, M., & Lenci, A. (2011). How we BLESSed distributional semantic evaluation. Edinburgh, Scotland: Proceedings of the GEMS 2011 - Workshop on Geometrical Models of Natural Language Semantics (EMNLP, 2011).

Baroni, M., Bernardi, R., Do, N.-Q., & Shan, C.-C. (2012). Entailment above the word level in distributional semantics. Avignon, France: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics,.

Baroni, M., Bernardini, S., Ferraresi, A., & Zanchetta, E. (2009). The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation Journal, 43*(3), 209-226.

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. Proceedings of Association for Computational Linguistics (ACL, 2014).

Barsalou, L. W. (2008). Grounded Cognition. *Annual Review of Psychology, 59*, 617-645.

Benotto, G. (2015). *Distributional Models for Semantic Relations: A Study on Hyponymy and Antonymy*. Pisa, Italy: PhD Thesis, University of Pisa.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32.

Bruni, E., Boleda, G., Baroni, M., & Tran, N.-K. (2012). Distributional semantics in technicolor. Jeju Island, Korea: Proceedings of the Association for Computational Linguistics (ACL, 2012).

Bruni, E., Tran, N. K., & Baroni, M. (2013). Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research, 48*.

Bullinaria, J., & Levy, J. (2012). Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior Research Methods, 44*(3), 890-907.

Casagrande, J. B., & Hale, K. L. (1967). Semantic relationships in Papago folk-definitions. In D. H. (eds.), *Studies in Southwestern ethnolinguistics* (p. 165-196). The Hague: Mouton.

Chersoni, E., Santus, E., Lenci, A., Blache, P., & Huang, C.-R. (2016). Representing Verbs with Rich Contexts: an Evaluation on Verb Similarity. Austin, Texas (USA): In Proceedings of Empirical Methods on Natural Language Processing (EMNLP, 2016).

Chersoni, E., Rambelli, G., Santus, E. (2016b). CogALex-V Shared Task: ROOT18. Osaka, Japan: Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V).

Chierchia, G. (1997). *Semantica*. Bologna, Italy: Il Mulino.

Church, K. W., & Hanks, P. (1989). Word Association Norms, Mutual Information and Lexicography. Vancouver, Canada: Association for Computational Linguistics (ACL, 1989).

Clark, S., & Pulman, S. (2007). Combining Symbolic and Distributional Models of Meaning. Stanford, CA: Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium on Quantum Interaction.

Clarke, D. (2009). Context-theoretic semantics for natural language: an overview. Athens, Greece: Proceedings of EACL 2009 Workshop on GEMS: GEometrical Models of Natural Language Semantics.

Collins, A. M., & Loftus, E. F. (1975). A spreading activation of semantic processing. *Psychological Review, 82*, 407-428.

Collins, A. M., & Quillian, R. M. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior, 8,* 240-247.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. Helsinki, Finland: International Conference on Machine Learning (ICML, 2008).

Conrad, C. (1972). Cognitive economy in semantic memory. *Journal of Experimental Psychology*, *92*(2), 149-154.

Cruse, D. A. (1986). *Lexical Semantics.* Cambridge, UK: Cambridge University Press.

Cruse, D. A. (2000). *Meaning in Language: An Introduction to Semantics and Pragmatics.* Oxford, UK: OUP.

de Marneffe, M-C., Rafferty A., & Manning C. D. (2008). Finding contradictions in text. Columbus, Oh: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08).

Dell'Orletta, F. (2009). Ensemble System for Part-of-Speech Tagging. Proceedings of EVALITA 9.

Deza, M. M., & Deza, E. (2009). *Encyclopedia of Distances.* Berlin, Germany: Springer.

Ding, J., & Huang, C.-R. (2013). Markedness of Opposite. In P. L. (eds), *Chinese Lexical Semantics* (p. 191-195). Heidelberg, Germany: Springer, Heidelberg.

Evert, S. (2005). *The Statistics of Word Cooccurrences: Word Pairs and Collocations* (Dissertation ed.). Stuttgart, Germany: Institut für maschinelle Sprachverarbeitung, University of Stuttgart.

Fellbaum, C. (1995). Co-occurrence and antonymy. *International Journal of Lexicography, 8*, 281-303.

Fellbaum, C. (1998). *WordNet: An electronic lexical.* Cambridge, MA: MIT Press.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2001). Placing search in context: The concept revisited. Hong Kong: Proceedings of the 10th International Conference on World Wide Web.

Firth, J. R. (1957). *Papers in Linguistics.* London, UK: Oxford University Press.

Geffet, M. Z., & Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. Ann Arbor, MI: Proceedings of ACL 2005.

Georges, T. M. (2003). *Digital Soul: Intelligent Machines and Human Values.* Boulder, CO: Westview Press Inc.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors. *Biological Cybernetics, 23*, 121-134.

Harris, Z. (1954). Distributional structure. *Word*, *10(23)*, 146-162.

Hearst, M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. Nantes, France: Fourteenth International Conference on Computational Linguistics (COLING, 1992).

Hebb, D. (1949). *The Organization of Behavior.* New York: Wiley & Sons.

Herbelot, A., & Ganesalingam, M. (2013). Measuring semantic content in distributional vectors. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL, 2013).

Herger, P. (2014). *Learning Semantic Relations with Distributional Similarity.* Berlin, Germany: Master's Thesis - Technische Universitat Berlin.

Herrmann, C., & Herrmann, D. J. (1984). The similarity and diversity of semantic relations. *Memory and Cognition, 12*, 134-141.

Hill, F., Roi, R., & Korhonen, A. (2014). SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *Computational Linguistics*.

Hirst, G. (2009). Ontology and Lexicon. In *Handbook on ontologies* (p. 269-292). Berlin, Heidelberg: Springer.

Hopfield. (1982). Neural networks and physical systems with emergent collective computational abilities. (p. 2554-2558). Proceedings of the National Academy of Sciences.

Huang, C.-R., Su, I.-L., Hsiao, P.-Y., & Ke, X.-L. (2007). Paranyms, Co-Hyponyms and Antonyms: Representing Semantic Fields with Lexical Semantic Relations. Hong Kong: Proceedings of Chinese Lexical Semantics Workshop 2007.

Huang, E. H., Manning, C. D., & Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. Jeju Island, Korea: Proceedings of the Association for Computational Linguistics (ACL, 2012).

Hull, D. (1996). Stemming algorithms – a case study for detailed evaluation. *Journal of the American Society for Information Science, 47*(1).

Jarmasz, M., & Szpakowicz, S. (2003). Roget's thesaurus and semantic similarity. Borovets, Bulgaria: Proceedings of RANLP 2003.

Johnson-Laird, P. N., Herrmann, D. J., & Chaffin, R. (1984). Only connections: A critique of semantic networks. *Psychological Bulletin, 96*(2), 292-315.

Jones, M. N., Willits, J., & Dennis, S. (2015). Models of Semantic Memory. In J. R. (Eds.), *Oxford Handbook of Mathematical and Computational Psychology* (p. 232-254).

Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing (2nd edition).* Prentice-Hall.

Justeson, J. S., & Katz, S. M. (1991). Co-occurrences of antonymous adjectives and their contexts. *Computational Linguistics, 17*, 1-19.

Katz, J. J., & Fodor, J. A. (1963). The Structure of a Semantic Theory. *Language, 39*(2), 170-210.

Kempson, R. M. (1977). *Semantic Theory*. Cambridge, UK: Cambridge University Press.

Kohonen, T. (1982). Self-organized formation of topologically. *Biological Cybernetics, 43*, 59-69.

Kotlerman, L., Dagan, I., Szpektor, I., & Zhitomirsky-Geffet, M. (2010). Directional Distributional Similarity for Lexical Inference. *Special Issue of Natural Language Engineering on Distributional Lexical Semantics. Natural Language Engineering, 16*(4), 359-389.

Kraaij, W., & Pohlmann, R. (1996). Viewing Stemming as Recall Enhancement. Proceedings of SIGIR96.

Landau, B., & Gleitman, L. R. (1985). *Language and Experience: Evidence from the Blind Child.* Cambridge, MA: Harvard University Press.

Landauer, T., & Dumais, S. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104*(2), 211-240.

Lenci, A. (2008). Distributional approaches in linguistic and cognitive research. *Italian Journal of Linguistics, 1(20),* 1-31.

Lenci, A. (2010). The life cycle of knowledge. In , *Ontology and the Lexicon. A Natural Language Processing Perspective* (p. 241-257). Cambridge, UK: Cambridge University Press.

Lenci, A., & Benotto, G. (2012). Identifying hypernyms in distributional semantic spaces. Montreal, Canada: First Joint Conference on Lexical and Computational Semantics (*SEM).

Lenci, A., Baroni, M., Cazzolli, G., & Marotta, G. (2013). BLIND: a set of semantic feature norms from the congenitally blind. *Behavior Research Methods, 45*(4), 1218-1233.

Levy, O., Goldberg, Y., & Dagan, I. (2015a). Improving Distributional Similarity with Lessons Learned from Word Embeddings. *TACL*.

Levy, O., Remus, S., Biemann, C., & Dagan, I. (2015b). Do Supervised Distributional Methods Really Learn Lexical Inference Relations? Denver, Colorado: In Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL, 2015).

Lewis, D. Y. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research, 5*.

Liu, H., Neergaard, K., Santus, E., & Huang, C.-R. (2016). EVALution-MAN: A Chinese Dataset for the Training and Evaluation of DSMs. Portorož, Slovenia: Proceedings of Language Resources and Evaluation Conference (LREC, 2016).

Lobanova, A. (2012). *The Anatomy of Antonymy: a Corpus-driven Approach.* Groningen, Netherlands: Dissertation. University of Groningen.

Lobanova, A., Kleij, T. v., & Spenader, J. (2010). Defining antonymy: A corpus-based study of opposites by lexico-syntactic patterns. *International Journal of Lexicography, 23*(1), 19–53.

Lowe, W. (2001). Towards a theory of semantic space. Edinburgh, UK: Proceedings of CogSci.

Lucerto, C., Pinto, D., & Jiménez-Salazar, H. (2002). An automatic method to identify antonymy. Workshop on Lexical Resources and the Web for Word Sense Disambiguation.

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, 28,* 203-208.

Lyons, J. (1977). *Semantics (2 Vols.).* Cambridge: Cambridge University Press.

Manning, C., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing.* Cambridge, MA: MIT Press.

Markman, E. (1981). Two different principles of conceptual organization. In A. B. & M.E. Lamb (eds.), *Advances in developmentalpsychology*. Hillsdale, NJ: Erlbaum.

Marton, Y. A. (2011). Filtering antonymous, trend-contrasting, and polarity-dissimilar distributional paraphrases for improving statistical machine translation. Proceedings of the Sixth Workshop on Statistical Machine Translation.

Matthews, P. (1991). *Morphology, 2nd ed.* Cambridge: Cambridge University Press.

McClelland, J. L., & Rumelhart, D. E. (1986). *Parallel Distributed Processing. Explorations in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models* (Vol. 2). Cambridge, MA: MIT Press.

McLeod, P., Shallice, T., & Plaut, D. C. (2000). Visual and semantic influences in word recognition: Converging evidence from acquired dyslexic patients, normal subjects, and a computational model. *Cognition, 74,* 91-114.

Mihalcea, R. a. (2005). Making computers laugh: Investigations in automatic humor recognition. Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing.

Mikolov, T., Yih, W.T., & Geoffrey, Z. (2013). Linguistic Regularities in Continuous Space Word Representations. 746-751: Proceedings of North American Association for Computational Linguistics (NAACL-HLT, 2013).

Miller, G. (1998). Nouns in wordnet. In C. Fellbaum (eds.), *WordNet: An Electronic Lexical Database* (p. 23-46). Cambridge, MA: MIT Press.

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes, 6*(1), 1-28.

Mohammad, S., Dorr, B., & Hirst, G. (2008). Computing word-pair antonymy. Waikiki, Hi: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2008).

Mohammad, S., Dorr, B., Hirst, G., & Turney, P. D. (2013). Computing lexical contrast. *Computational Linguistics*, *39*(3), 555-590.

Murphy, G. L. (2002). *The Big Book of Concepts.* Cambridge, MA: MIT Press.

Murphy, G. L., & Medin, D. L. (1985). The Role of Theories in Conceptual Coherence. *Psychological Review, 92*(3).

Murphy, M. L. (2003). *Semantic Relations and the Lexicon: Antonymy, Synonymy and other Paradigms.* Cambridge University Press.

Nakov, P., & Kozareva, Z. (2011). Combining Relational and Attributional Similarity for Semantic Relation Classification. Bulgaria: Proceedings of the Conference on Recent Advances in Natural Laguage Processing (RANLP 2011).

Ogden, C. K., & Richards, I. A. (1923). *The meaning of meaning.* London, UK: Routledge and Keegan Paul.

Osgood, C. E. (1971). Exploration in semantic space: A personal diary. *Journal of Social Issues, 27*, 5-64.

Ouyang, L., Boroditsky, L., & Frank, M. C. (2016). Semantic coherence facilitates distributional learning. Proceedings of the 34th Annual Meeting of the Cognitive Science Society.

Padó, S., & Lapata, M. (2007). Dependency-based Construction of Semantic Space Models. *Computational Linguistics, 33*(2), 161-199.

Pennacchiotti, M., & Pantel, P. (2006). Ontologizing semantic relations. Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics.

Pustejovsky, J. (1995). *The Generative Lexicon.* Cambridge, MA: MIT Press.

Rimmel, L. (2014). Distributional Lexical Entailment by Topic Coherence. Gothenburg, Sweden: Proceedings of EACL 2014.

Rips, L. J., Shoben, E. J., & Smith, F. E. (1973). Semantic distance and the verification of semantic relations. *Journal of Verbal Learning and Verbal Behavior*, *14*, 665-681.

Rogers, T. T., & McClelland, J. L. (2006). *Semantic Cognition.* Cambridge, MA: MIT Press.

Roller, S., Erk, K., & Boleda, G. (2014). Inclusive yet Selective: Supervised Distributional Hypernymy Detection. Dublin, Ireland: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers.

Roth, M., & Schulte im Walde, S. (2014). Combining word patterns and discourse markers for paradigmatic relation classification. Baltimore: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL).

Rumelhart, D. E., & Todd, P. M. (1993). Learning and connectionist representations. In D. E. (eds.), *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience* (p. 3-30). Cambridge, MA: MIT Press.

Sahlgren, M. (2006). *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.* Stockholm, Sweden: Ph.D. dissertation, Department of Linguistics, Stockholm University.

Salton, G. M., Yang, C. S., & Yu, C. T. (1975). A Theory of Term Importance in Automatic Text Analysis. *Journal of the American Society for Information Science*, *26*(1).

Santus, E., Chersoni, E., Lenci, A., Huang, C.-R., & Blache, P. (2016a). Testing APSyn against Vector Cosine on Similarity Estimation. Seoul, South Korea: Proceedings of Pacific Asia Conference on Language, Information and Computation (PACLIC, 2016).

Santus, E., Chiu, T.-S., Lu, Q., Lenci, A., & Huang, C.-R. (2016b). Unsupervised Measure of Word Similarity: How to Outperform Co-occurrence and Vector

Cosine in VSMs. Phoenix, Arizona: Proceedings of American Association for the Advancement of Artificial Intelligence (AAAI, 2016).

Santus, E., Chiu, T.-S., Lu, Q., Lenci, A., & Huang, C.-R. (2016c). What a Nerd! Beating Students and Vector Cosine in the ESL and TOEFL Datasets. Portorož, Slovenia: Proceedings of 10th Conference on Language Resources and Evaluation (LREC, 2016).

Santus, E., Gladkova, A., Evert, S., & Lenci, A. (2016d). The CogALex-V shared task on the corpus-based identification of semantic relations. Osaka, Japan: Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V), pages 69–79.

Santus, E., Lenci, A., Chiu, T.-S., Lu, Q., & Huang, C.-R. (2016e). Nine Features in a Random Forest to Learn Taxonomical Semantic Relations. Portorož, Slovenia: Proceedings of Language Resources and Evaluation Conference (LREC 2016).

Santus, E., Lenci, A., Chiu, T.-S., Lu, Q., & Huang, C.-R. (2016f). ROOT13: Spotting Hypernyms, Co-Hyponyms and Randoms. Phoenix, Arizona: Proceedings of Association for the Advancement of Artificial Intelligence (AAAI).

Santus, E., Lenci, A., Lu, Q., & Huang, C.-R. (2015a). When Similarity Becomes Opposition: Synonyms and Antonyms Discrimination in DSMs. *Italian Journal on Computational Linguistics*.

Santus, E., Yung, F., Lenci, A., & Huang, C.-R. (2015b). EVALution 1.0: an Evolving Semantic Dataset for Training and Evaluation of Distributional Semantic Models. Beijing, China: Proceedings of The 4th Workshop on Linked Data in Linguistics (LDL-2015), (ACL, 2015).

Santus, E., Lenci, A., Lu, Q., & Schulte im Walde, S. (2014a). Chasing Hypernyms in Vector Spaces with Entropy. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014), 2* (p. 38-42).

Santus, E., Lu, Q., Lenci, A., & Huang, C.-R. (2014b). Taking Antonymy Mask off in Vector Space. Phuket, Thailand: Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation (PACLIC 2014).

Santus, E., Lu, Q., Lenci, A., & Huang, C.-R. (2014c). Unsupervised Antonym-Synonym Discrimination in Vector Space. Pisa, Italy: Atti della Conferenza di Linguistica Computazionale Italiana (CLIC-IT 2014).

Saussure, F. d. (1983). *Course in General Linguistics.* Open Court Publishing.

Scheible, S., & Schulte im Walde, S. (2014). A Database of Paradigmatic Semantic Relation Pairs for German Nouns, Verbs, and Adjectives. Dublin, Ireland: Proceedings of the International Conference on Computational Linguistics Workshop on Lexical and Grammatical Resources for Language Processing.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging using Decision Trees. Manchester, UK: Proceedings of International Conference on New Methods in Language Processing.

Schulte im Walde, S., & Köper, M. (2013). Pattern-based distinction of paradigmatic relations for German nouns, verbs, adjectives. In *Language Processing and Knowledge in the Web* (p. 184-198). Germany: Springer.

Schulte im Walde, S., & Melinger, A. (2008). An In-Depth Look into the Co-Occurrence Distribution of Semantic Associates. *Italian Journal of Linguistics, 20*(1), 89-128.

Searle, J. (1980). Minds, Brains and Programs. *Behavioral and Brain Sciences, 3*, 417-57.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal, 27*, 379-423 and 623-656.

Smith, E. E., Shoben, E. J., & Rips, L. J. (1974). Structure and process in semantic memory: A featural model for semantic decisions. *Psychological Review, 1*, 214-241.

Szpektor, I., & Dagan, I. (2008). Learning entailment rules for unary templates. Proceedings of International Conference on Computational Linguistics 2008.

Terra, E., & Clarke, C. (2003). Frequency estimates for statistical word similarity measures. Proceedings of HLT/NAACL 2003.

Tungthamthiti, P., Santus, E., Xu, H., Huang, C.-R., & Kiyoaki, S. (2015). Sentiment Analyzer with Rich Features for Ironic and Sarcastic Tweets. Shanghai, China: Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC, 2015).

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, *49*, 433-460.

Turney, P. D. (2001). Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. Freiburg, Germany: Proceedings of the Twelfth European Conference on Machine Learning (ECML, 2001).

Turney, P. D. (2008). A uniform approach to analogies, synonyms, antonyms and associations. Manchester, UK: Proceedings of the 22nd International conference on Computational Linguistics (COLING, 2008).

Turney, P. D., & Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Articial Intelligence Research, 37*, 141-188.

Tversky, A. (1977). Features of similarity. *Psychological Review, 84*(4), 327-352.

Vigliocco, G., & Vinson, D. P. (2007). Semantic representation. In *The Oxford handbook of psycholinguistics* (p. 195-215). London, UK: University of London.

Weeds, J., & Weir, D. (2003). A General Framework for Distributional Similarity. Sapporo, Japan: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003).

Weeds, J., Clarke, D., Reffin, J., Weir, D., & Keller, B. (2014). Learning to Distinguish Hypernyms and Co-Hyponyms. Dublin, Ireland: Proceedings of International Conference on Computational Linguistics: Technical Papers (COLING, 2014).

Weeds, J., Weir, D., & McCarthy, D. (2004). Characterising measures of lexical distributional similarity. Geneva, Switzerland: Proceedings of International Conference on Computational Linguistics 2004.

Weizenbaum, J. (1966). ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the Association for Computing Machinery, 9*, 36-45.

Witten, I. H., Frank, E., & Hall, M. A. (2005). *Data Mining: Practical machine learning tools and techniques* (2nd ed.). Morgan Kaufmann.

Xu, H., Santus, E., Laszlo, A., & Huang, C.-R. (2015). LLT-PolyU: Identifying Sentiment Intensity in Ironic Tweets. Denver, Colorado: Proceedings of the 9th Workshop on Semantic Evaluation (SemEval-2015).