UNIVERSITÀ DI PISA

Master's Degree Course in Humanities Computing

Distributional methods for sentiment analysis

Candidate:
Emmanuele Chersoni

Supervisor:
Prof. Alessandro Lenci

Co-supervisor 1:
Prof. Dino Pedreschi

Co-supervisor 2:
Prof. Mirko Tavoni

Department of Philology, Literature and Linguistics

Academic Year 2012-2013

Table of contents

1

Sentiment Analysis: an introductive view

Introduction

In the last years, the Web has made available an incredible amount of user-generated content, which also means a huge availability of opinions. Opinions give us information about how reality is perceived by other people; consequently, they have a great influence on our thoughts, feelings and choices.

On the one hand, companies and industries are interested in opinions on the Web, because they can check their public image among the consumers and perform market surveys about their products and the ones of the competitors, in order to plan their market strategies. On the other hand, political organizations are trying to extract meaningful information from the mass of opinions expressed by the citizens in the new social media, so that they can achieve a real-time understanding of people's concerns. [1]

The area of study dealing with computational treatment of opinion,

[1]For a complete overview on these topics, see B. Liu, *Sentiment Analysis and Opinion Mining*, in *Synthesis Lectures on Human Language Technologies*, edited by G. Hirst, Morgan and Claypool Publishers, 2012; Y. Mejova, *Sentiment analysis: an overview*, Comprehensive exam paper, University of Iowa, 2009 (available at http://www.cs.uiowa.edu/~ymejova/publications/CompsYelenaMejova.pdf); B. Pang, L. Lee, *Opinion mining and sentiment analysis*, in *Foundations and trends in Information Retrieval,* vol. 2, n. 1-2, 2008, pp. 1-135; .A. Esuli, *Automatic generation of Lexical Resources for Opinion Mining: models, algorithms and applications,* PhD Thesis, PhD School on Information Engineering "Leonardo da Vinci" (supervisors: F. Sebastiani, L. Simoncini), University of Pisa, 2008.

sentiment and subjectivity has been called in many different ways (more or less equivalent) : *opinion mining* and *sentiment analysis* are the most common expressions. Probably, the focus of the attention in these two expressions is not the same: the research identified as *opinion mining* aims at building tools that can "process a set of search results for a given item, generating a list of product attributes... and aggregating opinions about each of them" [2]. The expression *sentiment analysis* is perhaps more concerning the specific application of classifying reviews according to their polarity (either positive or negative).

Nevertheless, *opinion mining* and *sentiment analysis* denote the same field of study, so they can be used interchangeably. Since the focus of my work is on the task of determining the polarity of the terms, I prefer to use the latter expression.

1.1 The tasks of Sentiment Analysis

Sentiment Analysis deals with complex problems which must be analysed distinctly, for this area of study encompasses separate tasks:

- **sentiment detection**, the classification of a text as objective or subjective, usually carried out by analyzing opinion-bearing words in the sentences;
- **polarity classification**, the classification of an opinionated

[2]K. Dave, S. Lawrence, D.M. Pennock, *Mining the peanut gallery: opinion extraction and semantic classification of product reviews*, in *Proceedings of the 12th International Conference on the World Wide Web*, Budapest, 2003, pp. 519-528.

piece of text as positive or negative (or, in alternative, its collocation on the continuum between these two extremes). In the most simple case, this can be seen as a binary classification task;

- **score assignment on a multi-point scale**, which is often used in the classification of product reviews. This task is very similar to a multi-class text categorization one, with the crucial difference that the vocabularies for each class, unlike the topic-based classification problem, can be very much alike and differ only in few crucial words (for example, the words indicating negation).

These are probably the three main tasks in Sentiment Analysis. But there are still two more tasks that it is worth mentioning:

- **discovery of the opinion's target**, clearly a task performed on texts where the target is not predefined, as in the case of the review of a product. Webpages and blogs often do not have a predefined topic and mention many objects; but still we have to remember that also a review of a product can be seen as mentioning several features of the same object, which can be seen as different opinions about different targets;
- **feature extraction**, the automatic recognition of components or attributes of an object, which can bring us to a more refined analysis of the sentiments. [3]

Finally, even if it is not -strictly speaking- a task dealing with the extraction of knowledge from texts, the **opinion-oriented summarization** of a document (or a set of) is a crucial step: we have to represent the sentiment information we have extracted in a

[3] Y. Mejova, *ibid.*, pp. 6-8.

way that is intuitive, easily interpretable and sufficiently informative. [4]

Summaries can be textual or non-textual and they can make use of graphs and/or charts. Obviously, when we have to summarize a lot of opinions, we have to represent only the information which is relevant for our goal.

1.2 Applications

While the World Wide Web is growing at an incredible rate, an increasing amount of user-generated content becomes available. In blogs, forums, customer reviews and social networks, Web users produce a huge quantity of subjective text. The feedback of consumers is important for business, because it enables to plan marketing strategies based on the consumers' reception. On the one side, Sentiment Analysis can help to handle a negative reception of a product, through the automatical extraction of the opinions about its distinctive features and -consequently- of the reasons why they are being criticised; on the other side, positive reviews of a product have a very positive impact on its sales, because a lot of Web users take the experience of other consumers as a reference for their future purchase. [5]

Furthermore, as Pang and Lee have stressed, a system that is able to

[4]B. Pang, L. Lee, *ibid.*, pp. 37-54.

[5]D. Lee, O. Jeong, S. Lee, *Opinion mining of customer feedback data on the Web,* in *Proceedings of the 2nd international conference on Ubiquitous information management and communication ICUIMC,* Association for Computing Machinery, New York, 2008, pp. 230-235.

find reviews and opinion expression on the Web and to create condensed versions of individual reviews, or a digest of overall consensus points, could be a precious saving-time resource, dispensing the data analysts from reading hundreds of similar judgements. [6] Government intelligence is an important field of application too. Traditionally, the only way to collect people's feedback about a government decision in a structured manner are *ad hoc* surveys, which are still very expensive in terms of time and money, and often ineffective because people are not always interested in answering surveys. Finally, they detect "known problems" through predefined questions and interviews, failing to detect "the unexpected". Opinion mining tools deal with unstructured text data, freely provided by Web users: they don not have the problem of "the predefined answer", and it is not even necessary to spend time to prepare surveys that people could not consider. These tools are gradually becoming more important in political life, since politicians have started to monitor public opinion in social media to understand public reaction to their position.Business and government intelligence are actually the main and widest areas of application of opinion mining / sentiment analysis. For example, they have also been used to deal with discussions about legal matters in weblogs [7], or with sociological problems like the circulation and diffusion of ideas. [8]

[6]See B. Pang, L. Lee, *ibid.*, p. 8-9.
[7]J. G. Conrad, F. Schilder, *Opinion mining in legal blogs,* in *Proceedings of the International Conference on Artificial Intelligence and Law (ICAIL),* New York, 2007, pp. 231-236.
[8]A. Kale *et al.*, *Modeling trust and influence in the blogosphere using link polarity*, in *Proceedings of the International Conference on Weblogs and Social Media (ICWSM),* Boulder (Colorado), 2007.

1.3 Methodologies

In this section, we will try to describe the most common techniques used in the field of Sentiment Analysis. [9]

A lot of tasks in Sentiment Analysis can be thought (as we said in section 1.1) of as problems of classification: usually, the methods of automatic text classification convert a piece of text into a feature vector that makes its most important features available, i.e. the features of that piece of text that are most salient for the task. Consequently, the main question concerns the features needed.

Let's summarize briefly the typologies of features commonly used in Sentiment Analysis:

- **term presence and frequency.** The TF-IDF measure, one of the most used in information retrieval, is based on the idea that terms appearing in the document, but rarely in the rest of the collection, are the most useful to understand the topic of the document. [10] Unlike traditional information retrieval, in Sentiment Analysis the raw presence of the term seems to be more significative, since Pang and Lee improved the performance of their system by using a boolean feature indicating term presence instead of frequency. [11]

  According to Wiebe, people are more creative when they are

[9]The main references for this section are A. Esuli, *ibid.*, 2008, pp. 5-11; Y. Mejova, *ibid.*, pp. 8-20.

[10]K. S. Jones, *A statistical interpretation of term specificity and its application in retrieval*, Journal of Application, Emerald, vol. 28, no. 1, 1972.

[11]See B. Pang, L. Lee, S. Vaithyanathan, *Thumbs up? Sentiment Classification using machine learning techniques*, *Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, Philadelphia, 2002, pp. 79-86.

being opinionated [12]: the consequence of this creativity could be the increased importance of rare terms in opinionated texts;

- **n-grams**, that is to say units composed of *n* word-stem, part-of-speech pairs (for example, ***in***-*prep* ***the***-*det* ***kitchen***-*noun* is a 3-gram). N-grams are useful because they encode information about terms' position, which can be crucial in determining the polarity of a phrase; [13]
- **part-of-speech**, adjectives are commonly exploited in Sentiment Analysis, because they're good indicators of sentiment in text. For instance, part-of-speech patterns, most of them including adjectives or adverbs, have been used by Turney for sentiment detection at the document level; [14]
- **syntactic information**, including text features such as negations, intensifiers and diminishers; [15]
- **negations** have an important role in Sentiment Analysis, since sentences having a very similar representation in a bag-of-words model can have an opposite polarity because of a single negation word. Negations can be handled in the post-processing of results (that's the way chosen by Hu and Liu

[12]J. M. Wiebe *et al.*, *Learning subjective language*, Computational Linguistics, MIT Press Journals, vol. 30, n. 3, 2004.
[13]N-grams are used, for instance, in J. M. Wiebe *et al., ibid.*, 2004.
[14]See P. D. Turney, *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*, in *Proceedings of the 40 th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, 2002, pp. 417-424.
[15]For an example of an approach using this kind of information, see A. Kennedy, D. Inkpen, *Sentiment Classification of Movie Reviews Using Contextual Valence Shifters*, Journal of Computational Intelligence, vol. 22, n. 2, 2006, pp. 110-125.

[16]), or they can be directly included in the document representation by appending them to the terms closer to negation (the approach used, for instance, by Das and Chen[17]).

1.4 Identifying the semantic orientation

Another basic task in opinion mining is the **identification of semantic orientation of words**, namely their polarity. Examples of positive words are *good, beautiful, interesting,* while *bad, ugly, boring* are typical negative words There are various ways of identifying this polarity: we can use a lexicon, manually or automatically constructed; or we can exploit statistical information, like the co-occurrence of words with other words of a known polarity.

The most simple lexicon will include a list of words and their classification in 1) objective or subjective words, and in 2) positive or negative words. Further information could concern the *intensity* of the positive/negative meaning, that is to say the strength of the feeling associated with the word, and the *centrality*, namely the degree of relatedness to the category to which a word is assigned. Lexicons constructed in order to handle the ambiguity of the terms by assigning them to categories are said to be *fuzzy*: a lexical entry is

[16]M. Hu, B. Liu, *Mining and summarizing customer reviews*, in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP),* Vancouver, 2005.

[17]S. Das, M. Chen, *Yahoo! For Amazon: extracting market sentiment from stock message boards*, in *Proceedings of the 8th Asia Pacific Finance Association Annual Conference*, Bangkok, 2001.

not assigned to a single class, corresponding to its orientation, but to multiple classes and it has different scores of relatedness to each class. [18]

As we previously anticipated, sentiment lexicons can be created through manual annotation. But, of course, there are other ways: for example, exploiting the availability of a resource like WordNet [19], Esuli and Sebastiani expanded it by adding polarity and objectivity labels for each term. They assigned a label to every *synset* of Wordnet (a *synset* is, essentially, a group of synonyms) by using a set of ternary classifiers, each able to decide whether a synset is positive, negative or objective. [20]
An example of a SentiWordNet synset:

NOUN

unhappiness sadness P: 0 O: 0.25 N: 0,75
emotion experienced when not in a state of well-being

sorrowfulness sorrow sadness P: 0 O: 0.375 N: 0.625
the state of being sad: "she tired of his perpetual sadness"

[18]P. Subasic, A. Huettner, *Affect analysis of text using fuzzy semantic typing*, Institut of Electric and Electronics and Engineering – Finland Section, vol. 9, n. 4, pp. 483-496, 2001.
[19]The database *WordNet* is available at the URL http://wordnet.princeton.edu/wordnet/download/. For a complete description of its story and its features, see: C. Fessbaum, *Wordnet: an electronic lexical database*, MIT Press, Cambridge, 1998. See also: G. A. Miller, *Wordnet: a lexical database for English*, Communication of the Association for Computing Machinery, vol. 38, n. 11, 1995, pp. 39-41; C. Fellbaum, *WordNet and wordnets,* in K. Brown *et alii, Encyclopedia of Language and Linguistics*, Elsevier, Oxford, 2005, pp. 665-670.
[20]See A. Esuli, F. Sebastiani, *SentiWordNet: a publicly available lexical resource for Opinion Mining*, in *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC 2006),* Genova, 2006, pp. 417-422; A. Esuli, *ibid.*, 2008.

sadness lugubriousness gloominess P: 0 O: 0.125 N: 0.875

the quality of excessive mournfulness and uncheerfulness

In particular, Esuli and Sebastiani exploited WordNet's semantic structure, namely the lexical relations between synsets: they started from two seed sets of words of a known polarity and used the lexical relations defined in WordNet to find new terms which can be considered representative of the two categories (for instance: if a synset is the antonym of another one ,whose polarity is known, its own polarity will be probably the opposite one). Then, all the glosses associated to a term are collated, so as to form a textual representation of its "definition"; every term representation is then converted into a vectorial form. In the training phase, every term of the seed set are used as training examples of the categories; then, finally, the resulting ternary classifier is applied to the vectorial representations of all WordNet synsets (to the Objective category are assigned the terms which are neither positive, nor negative). [21] Every synset has assigned three scores, corresponding to the three possible polarities, and their sum is 1.0, in order to give a graded evaluation of the opinion-related properties of every term.

Kamps *et alii*, instead of using seed sets, focused on the structure of the graph defined by WordNet's lexical relations: they took, as nodes of their graph, all the adjectives contained in the intersection between

[21]A. Esuli, *ibid.*, pp. 11-21, 2008.

WordNet and their term sets, and they added an edge between two adjectives if they had a synonymy relation. Then, they defined a *geodesic distance d (t1, t2)* between terms *t1* and *t2*, corresponding to the length of the shortest path connecting *t1* and *t2*. The orientation of a term is determined by calculating its relative geodesic distance from the basic terms **good** and **bad**:

$$O(t) = (d(t, \textbf{bad}) - d(t, \textbf{good})) / d(\textbf{bad}, \textbf{good})$$

The adjective *t* is assigned to the **Positive** class if O (*t*) > 0, or to the **Negative** class if

O (*t*) < 0. Obviously, the higher is the absolute value of O (*t*), the stronger the semantic orientation of the adjective will be (the geodesic distance between **good** and **bad** serves as a normalization factor, in order to constrain the O values in the [-1,1] range). [22]

Other studies based on WordNet are those of Kim and Hovy and Hu and Liu, who aimed both at the generation of a vast lexicon of positive and negative terms, by starting from a small seed set of words of known polarities and expanding it through the exploitation of the antonymy and synonymy relations to determine their polarity assignment. [23]

In particular, Kim and Hovy's system starts from a set of positive and negative terms, and expand each set by adding to it the synonyms of its seed terms and the antonyms of the terms of the other seed set. A

[22]J. Kamps *et alii*, *Using WordNet to measure semantic orientation of adjectives*, in *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC-04)*, vol. IV, Lisbon, 2004, pp. 1115-1118.

[23]S. Kim and E. Hovy, *Determining the sentiment of opinions*, in *Proceedings of the 20th International Conference on Computational Linguistics,* Geneva (SUI), 2004, pp. 1-8; M. Hu and B. Liu, *ibid.*, 2005.

problem with this method is that it is limited to terms whose synonyms/antonyms are in the seed sets.

Also the approach of Turney and Littmann is based on the initial selection of two small sets of terms whose polarity is known to be positive or negative. [24] Their idea is to compute the *pointwise mutual information (PMI)* of every target word *w* with each seed set term t_i, in order to measure the degree of their semantic association. 25

The orientation of the word *w* was given by:

$$O(w) = \sum \text{PMI}(w, t(i)_{\text{POSITIVE}}) - \sum \text{PMI}(w, t(j)_{\text{NEGATIVE}})$$

that is to say, the sum of the scores of semantic association with the seed positive terms minus the sum of the scores of semantic association with the seed negative terms.

Turney and Littmann's assumption -a very important one, for our goal- is clearly that words tend to share the same semantic orientation of their neighbors. In the final part of their study, the two researchers also tested another method for computing the semantic association, namely the Latent Semantic Analysis: they applied the PMI-LSA measure only to the smallest of their document sets, because the computational cost of this technique was too high;

[24] P. Turney, M. Littman, *Measuring praise and criticism: inference of semantic orientation from association*, ACM Transactions on Information Systems, vol. 21, n. 4, 2003, pp. 315-346.

[25] The *pointwise mutual information* is a measure of semantic association between *t* and *w*, defined by Turney and Littman as:

PMI (t, w) = log p (*t*, *w*) / p (*t*) * p (*w*)

P (*t*) and p (*w*) indicate the probabilities of the single events *t* and *w*, while p *(t, w)* indicates the joint probability.

anyway, they observed notable improvements of the system's performance.

When the polarity of individual words has been individuated, it is often desirable to determine the polarity of larger textual units, like sentences and documents.

Hu and Liu simply chose to calculate the average of the polarity scores of the words in the sentence, so that they can assign the label corresponding to the dominating polarity [26].

Instead, Yu and Hatzivassiloglou trained a Naive Bayes classifier using sentences and documents labeled as instances of the categories of positive and negative: they used the presence of words of known polarities in a sentence to assign it a subjective label, and they also considered the effect of the negation words ("no", "not", "don't", "yet") which appeared in a window of 5 words around the target subjective word. [27]

Naturally, a more sophisticated computation of sentiment labels for textual units can be done by considering syntactic relationships between words. [28]

[26] See M. Hu, B. Liu, *ibid.*

[27] H. Yu, V. Hatzivassiloglou, *Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences*, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP),* Sapporo (JAP), 2003.

[28] See, for example, A. Popescu, O. Etzioni, *Extracting product features and opinions from reviews*, in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP),* Vancouver, 2005.

1.4 The problem of the contextual polarity

Until now, we have mostly considered the polarity of words taken in isolation. [29] As Osgood, Suci and Tannenbaum stated, it seems that individual words have a **prior polarity**, that is to say a semantic orientation indipendent of context; furthermore, the strength of these polarities can be expressed by a numerical value. [30] As we have said before, adjectives are commonly exploited in sentiment analysis, because they seem to be the primary source of subjective content in a text: the primary aim of a lot of studies is to individuate their prior polarity, assuming that the orientation of a whole sentence (and of more complex textual units) can be described by a value which results from some sort of combination of the adjectives and other relevant words' polarity values.

Obviously, noun and verbs can carry semantic polarity information too; Taboada and colleagues noticed that "*they often have both neutral and non-neutral connotations*" [31], in the sense that they tend to have more than a plausible interpretation in terms of polarity, so that contextual information is needed to disambiguate them. See for example the meaning of the verb *to inspire* in the following sentences:

1. *the teacher inspired her students to pursue their dreams;*

[29]For the following section, the main reference is M. Taboada *et alii*, *Lexicon-based methods for Sentiment analysis*, Computational Linguistics, MIT Press Cambridge, Boston, Vol. 37, n. 2, 2011, pp. 267-307.

[30]C. E. Osgood, G. Suci, P. Tannenbaum, *The measurement of meaning*, University of Illinois Press, Urbana (Illinois), 1957.

[31]M. Taboada *et alii*, *ibid.*, pp. 272-273. The following examples are brought from the same paper.

2. *this movie was inspired by true events.*

in the first sentence, the verb has a very positive meaning, while in the second one the meaning of *inspired* is rather neutral. Instead, adjectival polarity seems to be more stable across the possible contexts.

These facts clearly imply the existence of another kind of polarity, a **contextual polarity**, that is to say the polarity of a word in a particular linguistic context, where the meaning of the other linguistic elements selects only a portion of its potential meaning, discarding at the same time those interpretations that are not pertinent to the context. [32]

In an example like

3. He used an analgesic to kill the pain

two words having a negative prior polarity, *kill* and *pain*, combine in a way that a very specific meaning of the verb *kill* is activated, and the result is a positive phrase.

After these reflections, we could reconsider what we have generally called, until now, the *intensity* or the *strength* of the semantic

[32]The notion of "the meaning of a word", traditionally thought as a fixed unit of meaning associated to it, has been contested by cognitive semantics, among the others. According to Alan Cruse, a word has a semantic potential, i.e. it can be used to signify something within a certain region of the conceptual space, and every particular interpretation of the word is a point within that region. The sense units we retrieve, when we look for a word in our mental lexicon, are not ready *a priori*, but they're constructed at the moment and their delimitation is often determined by contextual factors.

See A. Cruse, *Meaning in language: an introduction to semantics and pragmatics*, Oxford Textbooks in Linguistics, Oxford University Press, 2004, particularly the chapter "Contextual variability in word meaning"; W. Croft, A. Cruse, *Cognitive linguistics*, Cambridge Textbooks in Linguistics, Cambridge University Press, 2004, particularly the chapter "Polysemy: the construal of sense boundaries".

orientation. Maybe, when we are asked to give a polarity and a score to a word, we think to some sort of prototypical context of that word, and our judgement refers to word in that particular context; and, probably, words obtaining higher polarity scores are those for which is more difficult to imagine a context where that polarity is reversed.

Other elements having an important influence on the polarity of the sentence are the intensifiers, the negators and the *irrealis*.

According to Quirk's classification, intensifiers can be divided in two major categories: *amplifiers*, which increase the strength of the polarity of a lexical item, and *downtoners*, which decrease it. [33]

4. The show was truly fantastic.

5. I'm feeling slightly tired after the walk

In the examples, the adverbs *truly* and *slightly* have -respectively- the effect of accentuating the positive judgement expressed in the first sentence and of attenuating the sensation described in the second one. Some researchers in sentiment-analysis have chosen to implement intensifiers by adding or subtracting a fixed value to the prior polarity of a neighbour element. [34]

This approach, valid perhaps for the majority of cases, has some limits, because the implementation of intensifiers should also consider variables like:

- the extent to which they modify the prior polarity;

[33]R. Quirk, *A comprehensive grammar of the English language*, Longman, London, 1985.
[34]See, for example L. Polanyi, A. Zaenen, *Contextual valence shifters*, in *Computing attitude and affect in text: theory and applications*, edited by J. Wiebe, Springer, Dordrecht, 2004, pp. 1-10; A. Kennedy, D. Inkpen, *ibid.*, 2006.

- the nature of the item being intensified, because -for instance- items that are already intense have a greater increase, when modified by an amplifier. For these reasons, Taboada *et alii* decided to model the process of polarity intensification by associating a percentage to every modifier, which is applied to the element modified's polar value and expresses the measure of its increase/decrease. [35]

Concerning the negation, the most simple approach is the reversal of the polarity of a lexical item next to a negator: *bad* (-3), for example, is changed into *not bad* (+3).

There are also negators which can occur at a significant distance from the lexical item they negate, and in these cases a backwards search is required.

Another important issue is whether a "polarity flip" is an efficient way to quantify the effects of the negation: for example, an adjective like *beautiful* (+5), when it is negated it is certainly less negative than *horrible* (-5). A possible solution to this problem, as suggested by Taboada *et alii*, is to shift the SO value towards the opposite polarity by a fixed amount, instead of reversing it. [36]

6. He is an excellent student (degree of positivity: 5).

7. He is not an excellent student (degree of positivity: 1)

In the examples, the first sentence has certainly a high degree of positivity; in the second sentence, it would not be right to reverse the polarity of the score of the positive sentence, because *not excellent* is

[35]M. Taboada *et alii*, *ibid.*, pp. 274-279.
[36]M. Taboada *et alii*, *ibid.*, pp. 274-279.

more positive than the opposite of *excellent*; consequently, instead of reversing the polarity, we shifted the SO value by 4 towards the opposite polarity.

Furthermore, negative items seems to interact with intensifiers in various and often unpredictable ways [37]: *not very beautiful*, for example, seems to be more negative than *not beautiful*, so that, if we want to preserve the notion of the polarity flip, we have to reverse the polarity of both the adjective and the intensifier. [38]

Finally, there are some markers indicating that words in a sentence might not be reliable for the aims of sentiment analysis: the term used by Taboada and colleagues to refer to these words is *irrealis*, usually applied in non-factual contexts. The English language has a few manners to convey *irrealis*, like modal and private-state verbs. See for example:

8. If he didn't stop playing basketball, he could have been one of the best point guards of the last decade.
9. I considered you a good friend (but now I've changed my mind).

Words like *best* and *good* have normally a positive value, but in these sentences there are some *irrealis* which make us understand that we're in a non-factual context (in the first sentence, the modal verb *could*; the verb *consider* in the second one).

Irrealis include also conditional markers (*if*), negative polarity items like *any* and *anything*, questions, words enclosed in quotes. A current

[37] A. Kennedy, D. Inkpen, *ibid.*, 2006.
[38] M. Taboada *et alii*, *ibid.*, pp. 274-279.

approach to handle them consists in ignoring the semantic orientation of all the words in the scope of an *irrealis* marker. [39]

1.5 Statistical semantics

Until now, it seemed necessary to make a clear distinction between prior polarity and contextual polarity, which is -if we want to formulate it in more general terms- the problem of the distinction between a pre-existing sense of a word, usually given by a dictionary, and the meaning the word has in a particular context.

The task of Word Sense Disambiguation can be seen as a choice, for a word that is met in a context, of a sense among those which can be found in a pre-compiled dictionary.

Interestingly, some of the researchers who have faced the problem of Word Sense Disambiguation have also questioned the traditional notion of "word sense". [40]

For example, Schütze and Pederson used high-dimensionality vectors to describe each occurrence of a target word, then they clustered the vectors. The better-defined of the resulting clusters corresponded, in their claim, to word senses, so that the Word Sense Disambiguation task could be seen as a comparison between the contextual vector of a new occurrence of the target word and the centroids of the clusters: they assigned the sense whose cluster centroid was closest to the occurrence vector.

In this way, word senses are not given *a priori*: they correspond to

[39]M. Taboada *et alii*, *ibid.*, pp. 274-279.

[40]See, for example, A. Kilgarriff, *I don't believe in word senses*, Computers and the Humanities, Springer, vol. 31, 1997, pp. 91-113.

contexts of occurrence of the word, which are clustered according to the criterion of their similarity. [41]

Results like those obtained by Schütze and Pederson could pose some interesting questions: if it is possible to individuate different word senses on the base of different distributional behaviours of the same target word, is it possible to extend this approach also to the problem of polarity assignment?

Furthermore: if our polarity judgements are dependent -as it seems- on the occurrences of the target word in some prototypical contexts, do a prior polarity *which is indipendent from its patterns of occurrence* really exist?

In an article published by Furnas *et al.* in 1983, statistical semantics was defined as follows: "*statistical semantics – studies of how the statistical patterns of human word usage can be used to figure out what people mean*" (Furnas *et al.*: 1983). [42]

The possibility of using those patterns for the study of meaning, generally enunciated there, underlies the Distributional Hypothesis: words that occur in similar contexts tend to have similar meanings. [43]

The idea of finding the evidence of the semantic properties of a word by inspecting its distributional and combinatorial behavior was introduced by Zellig Harris, whose proposal -aimed to guarantee the

[41]H. Schütze, J. Pederson, *Information retrieval based on word senses*, in *Proceedings of ACM Special Interest Group on Information Retrieval (SIGIR)*, 1995.
[42]G. Furnas *et al.*, *Statistical semantics: analysis of the potential performance of keyword information systems*, Bell System Technical Journal, vol. 62 , no. 6, 1983, pp. 1753-1806.
[43]The definition of these hypothesis is taken by P. Pantel, P. Turney, *From frequency to meaning: vector space models for semantics*, Journal of Artificial Intelligence Research, n. 37, 2010, pp. 141-188.

scientificity of the linguistic enterprise- was to evaluate the semantic similarity between linguistic expressions as a function of the degree of the similarity of the contexts in which they occur. [44]

While the distributional semantics inspired by Harris' work found several difficulties to stand out in theoretical linguistics, methods for distributional analysis of linguistic contexts are always been kept alive within the corpus linguistics tradition, as summarized by Firth's slogan "You shall know a word from the company it keeps". [45]

History, philosophy and applications of distributional semantics will be discussed extensively in the second chapter; now, in our perspective, it's worthwhile to ask ourselves if opinion-related properties of the meaning of a word can be studied on the same basis. We could start from a simple fact: our evaluations of entities, facts and actions can be thought as being distributed on an axis and comprised between two polar opposites, i.e. totally negative or totally positive; assumed that we are talking about a certain entity in a certain circumstance, it is highly probable that we will refer to it by using words whose collocations on the axis lie close.

So, if a word is carrying a *subjective content* and if we are able to recognize the "semantic orientation" of the context, we can reasonably speculate about its *polarity.* [46]

Our work hypothesis, which we are going to explore in the next chapters, could be formulated as follows: *words that occur in*

---

[44]Z. Harris, *Methods in structural linguistics*, University of Chicago Press, Chicago, 1951.
[45]J. Firth, *Papers in Linguistics 1934-1951*, Oxford University Press, London, 1957, p. 11.
[46]Even if we are concentrating on the problem of the polarity of a word, the task of recognizing the subjective/objective content of a word is anything but banal: see A. Esuli, F. Sebastiani, *Determining term subjectivity and term orientation for opinion mining*, in *Proceedings of EACL-06*, *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, 2006.

contexts with similar semantic orientations will tend to have similar polarity.

1.6 Sentiment lexicons and affective word lists

The final paragraphs of this chapter will be dedicated to a brief description of some publicly available resources -databases and lexicons- for sentiment analysis, which can provide us examples labeled with respect to their sentiment polarity, at different levels of granularity. [47]

The following list contains some English language sentiment lexicons (freely downloadable). It does not pretend to be exhaustive: we included only the lexicons which we had the possibility to examine, excluding -for example- word lists on which a documentation exists, but that are actually unavailable (because of technical problems, or for any other reason):

- **AFINN** is a list of English words rated for their valence with an integer between minus five (negative) and plus five (positive). The last version of the list contains 2477 words and phrases, manually labeled by Finn Arup Nielsen in 2009-2011. This lexicon has been projected for sentiment analysis in

[47] See B. Pang, L. Lee, *ibid.*, 2008, particularly the 7th chapter.

microblogs; [48]

- **General Inquirer**, a site which provides entry-points to resources associated with the General Inquirer. It's possible to find here lists of manually-classified terms with various kinds of markers (semantic orientation, cognitive orientation, mood of the speaker etc.); [49]
- **Hu-Liu list**, a list of positive and negative opinion words or sentiment words for the English language (around 6800 words) compiled over many years starting from Hu and Liu's first paper on this topic. There are also many misspelled words in the list and it's not a mistake: the authors decided to include them because they appear frequently in social media content; [50]
- **OpinionFinder's subjectivity lexicon**, this list of subjectivity clues is part of Opinion Finder and it's available for download. These clues were extracted from several sources; some of them were manually compiled, while others were identified automatically using both annotated and unannotated data. Examples of clues are the part of speech, the length of the clue in words, the prior polarity of the word (i.e. its polarity when considered out of context), the strength of the subjectivity the

[48] See F. A. Nielsen, *A new ANEW: Evaluation of a word list for sentiment analysis in microblogs*, *Proceedings of*
the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages 718, in CEUR
Workshop Proceedings : 93-98, 2011.
The word list is available at:http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010.
[49]http://www.wjh.harvard.edu/~inquirer/Home.html .
[50]The list is available at http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html; an example of its use can be found in the paper M. Hu, B. Liu, *ibid.*, 2005.

word expresses; [51]

- **SentiWordNet** is a lexical resource for Opinion Mining, in which three sentiment scores -positivity, negativity, neutrality- are assigned to each synset of WordNet through a semi-supervised approach, so that -for example- the positivity score associated to the synset *s* indicate how positive are the terms contained in *s* (see section 1.3).

Each of the scores ranges in the interval [0, 1] and their sum is 1 for each synset: this means that a synset may have non-zero scores for all the three "orientations", indicating that the terms of the synset have each of the three opinion-related properties to a certain degree; [52] according to a recent article, the number of SentiWordNet's single word entries is over 115000; [53]

- **WordNet Affect** is an extension of WordNet Domains, a lexical resource created in a semi-automatic way by augmenting WordNet with domain labels; the extension WordNet Affect consisted in the addition of another set of synsets representing affective concepts. The researchers assigned to a number of WordNet synsets one or more affective labels, called the a-labels; then, they added a new set of a-labels, hierarchically organized and modeled on the

[51]OpinionFinder's resources are available at http://www.cs.pitt.edu/mpqa/index.html. This lexicon was used in T. Wilson, J. Wiebe, P. Hoffmann , *Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis*, in *Proceedings of HLT-EMNLP,* 2005*,* pp. 347-354.

[52]SentiWordNet was introduced in the papers: A. Esuli, F. Sebastiani, *ibid*., 2006; F. Baccianella, A. Esuli, F. Sebastiani, *SentiWordNet 3.0: an enhanced lexical resource for Sentiment Analysis and Opinion Mining,* in *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC 2010),* Valletta (MT), 2010, pp. 2200-2204. The resource is downloadable at http://sentiwordnet.isti.cnr.it/.

[53]A. Das, S. Bandyopadhyay, *Towards the Global SentiWordNet*, *Proceedings of the 24th Pacific Asia Conference on Language Information and Computation 2010*, Tohoku University (Japan), 2010, pp. 799-808.

WordNet hyperonym-hyponym relation; finally, through the introduction of a-labels related to polarity (positive, negative, neutral, ambiguous), the synsets were distinguished also according to the emotional valence; [54]

- **NRC word-emotion association lexicon,** created by Turney and Mohammad. The lexicon has human annotations of emotion associations for more than 24,200 word senses (about 14,200 word types), and the annotations include whether the target is positive or negative, and whether the target has associations with eight basic emotions (joy, sadness, anger, fear, surprise, anticipation, trust, disgust). [55]

  To obtain a copy of the lexicon, it is sufficient to send an e-mail to saif.mohammad@nrc-cnrc.gc.ca;

- **Warriner and Kuperman's norms of valence, arousal and dominance**, which extended the ANEW norms to nearly 14 000 English lemmas, including the scores for the three components of valence (the pleasantness of the stimulus), arousal (the intensity of the emotion) and dominance (the degree of control exerted by the stimulus) and information on

[54]The description and the download of WordNet Affect are available at http://wndomains.fbk.eu/index.html.

WordNet Affect has been described in C. Strapparava, A. Valitutti, *Wordnet-affect: an affective extension of wordnet, i*n *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, 2004, pp. 1083-1086.

[55]The lexicon has been introduced in the following papers: S. Mohammad, P. D. Turney, *Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon*, In *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, Los Angeles (California), 2010; S. Mohammad, P. D. Turney, *Crowdsourcing a Word-Emotion Association Lexicon*, to appear in *Computational Intelligence*, Wiley Blackwell Publishing Ltd.

Further information can be found at http://www.umiacs.umd.edu/~saif/WebPages/ResearchInterests.html.

gender, age and educational differences in emotion norms. [56]

[56] A. Warriner *et al.*, *Norms of valence, arousal and dominance for 13915 English lemmas,* to appear in *Behaviour Research Methods.* Further information can be found at http://crr.ugent.be/archives/1003.

1.7 Datasets for Sentiment Analysis

In this section, we will see briefly the datasets and the corpora that could be used to build training and test sets for sentiment-related tasks.

The following list is in alphabetical order [57]:

- **Congressional Floor-Debate Transcripts**, a congressional-speech corpus, including a total of 3857 speech segments transcribed from 53 different debates. The main characteristics of this corpus are:
  - automatically derived labels for whether the speaker supported or opposed the legislation in debate;
  - informations about the debate from which each speech has been transcribed;
  - indications of by-name references between speakers, in order to allow experiments of agreement classification. [58]

The aim of the researchers who have built this corpus was to automatically determine the agreement or disagreement to the proposed legislation. Instead of classifiying speeches in isolation, they exploited their belonging to a wider discussion, so they studied the relationship between discourse segments in different speeches.

The downloadable file includes also the data of the graph of

[57] See B. Pang, L. Lee, *ibid.*, 2008, pp. 61-68.

[58] The dataset is available at http://www.cs.cornell.edu/home/llee/data/convote.html. It was introduced in the following article: M. Thomas, B. Pang, L. Lee, *Get out the vote: determining support or opposition from Congressional Floor-debate transcripts*, in *Proceedings of 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, 2006, pp. 327-355.

references between speech segments;

- **Cornell movie-review datasets,** these corpora were introduced in the articles of Pang and Lee for use in sentiment-analysis experiments. It is possible to download collections of movie-review with labels indicating the overall sentiment polarity (positive or negative) or subjective rating (the number of stars assigned) of the documents, for a total of 1000 positive and 1000 negative reviews, or collections of sentences labeled with respect to their subjectivity status, for a total of 5000 subjective and 5000 objective processed sentences; [59]

- **Customer review datasets,** a dataset consisting in the reviews of five electronic products dowloaded from Amazon and Cnet; recently an addendum, with nine products, has been made available. The researchers have focused themselves on the evaluation of features of the products, so they have labeled the opinion expressed on every particular feature with a positive or negative score; [60]

---

[59]The dataset is available at http://www.cs.cornell.edu/people/pabo/movie-review-data/.

Articles using this dataset are: B. Pang, L. Lee, S. Vaithyanathan, *ibid.* 2002; B. Pang, L. Lee, *A sentimental education : Sentiment Analysis using subjectivity summarization based on minimum cuts,* in *Proceedings of 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, Barcelona, 2004, pp. 271-278; B. Pang, L. Lee, *Seeing stars: exploiting class relationship for sentiment categorization with respect to rating scales*, in *Proceedings of 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP 2005),* Vancouver, 2005, pp. 115-124.

[60]The datasets are available at http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html. The basic version was introduced in M. Hu, B. Liu, *ibid.*, 2004.

- **MPQA Opinion Corpus**, a corpus containing 535 articles from a wide variety of news sources manually annotated for opinions and other private states (i.e. beliefs, emotions, sentiments etc.). The tool used for annotation is GATE, which is freely available from Sheffield University [61].
-

GATE organizes annotations in a document into different sets, and one of the sets of MPQA annotation is *expressivity-subjectivity.* Examples of tag attributes (and relative values) are: *polarity* (positive, negative, neutral, both), *intensity* (low, medium, high extreme), *es-uncertain* (somewhat-uncertain, very-uncertain; this attribute is used by the annotator to indicate that he is not sure of the expressivity-subjectivity of the word/phrase he's labeling); [62]

- **Multi-Domain Sentiment Dataset**, a dataset consisting of product reviews from four different product types (books, electronics, DVDs and kitchen appliances), with 1000 positive and 1000 negative reviews for each of these categories; in addition, the researchers added around 9000 instances of unlabeled data, in order to allow further experiments. Since the reviews have been taken from Amazon.com, they have a 1-to-5 star label: the reviews with a rating > 3 were considered to be positive, while those with rating < 3 were labeled negative.

[61]http://gate.ac.uk/download/

[62]The dataset is available at http://www.cs.pitt.edu/mpqa/mpqa_corpus.html. The introduction of the corpus, and the description of the annotation scheme, can be found in J. Wiebe *et al., Annotating expressions of opinions and emotions in language*, *Language Resources and Evaluation*, vol. 39, no. 2-3, 2005, pp. 165-210.

It is possible to download both the unlabeled version of the original dataset and the labeled one.[63]

[63]The datasets are available at http://www.cs.jhu.edu/~mdredze/datasets/sentiment/. The original version has been introduced in J. Blitzer, M. Dredze, F. Pereira, *Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification,* in *Proceeding of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, Prague, 2007, pp. 440-447.

2

From the distributional hypothesis to Vector Space Models for Sentiment Analysis

In the first chapter, we have presented an overview of Sentiment Analysis, of its applications and its methodologies; we have introduced the problems of prior and contextual polarity, together with a work hypothesis, i.e. studying the semantic orientation of linguistic items on distributional basis.

The distributional hypothesis is the ground on which rests a popular framework for the semantic representation, the Distributional Semantic Models (DSMs), which encode lexical meaning as high-dimensional vectors. The components of the vectors measure values of co-occurrence of the lemma with context features (they can be other words, or larger textual units, or documents). [64]

In this chapter, we intend to explain the origins of distributional semantics, its different "versions", its applications in computational semantics; finally, we will list some studies which have used a distributional approach to deal with sentiment-related problems.

[64]See K. Erk, S. Padó, *A structured vector space model for word meaning in context*, in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP – 08),* Honolulu, 2008, pp. 897-906.

2.1 What is distributional semantics?

The adjective *distributional* designates a wide range of approaches to semantics, characterised by a usage-based perspective on meaning and by the idea that the words' different semantic behaviours are correlated with different distributional behaviours. [65]

The focus of distributional methodology is on differences of meaning or, to express it differently, semantic similarities between linguistic items. Obviously, in such a perspective, we can investigate meaning only if we're able to specify under which conditions two linguistic items are semantically similar.

Distributional approaches all rely on some version of the Distributional Hypothesis:

> *The degree of semantic similarity between two linguistic expressions A and B is a function of the similarity of the linguistic contexts in which A and B can appear.*

Consequently, at least part of the meaning of a linguistic item can be inferred from its distributional properties, that is to say from the contexts in which it occurs.

Talking about the distributional hypothesis, it's easy to make reference to the post-bloomfieldian American structuralism, and in particular to the works of Zellig Harris.

Harris, who felt as necessary a solid methodological base for linguistic analyses, believed that members of the same linguistic

[65] To say it with Harris' words, "difference of meaning correlates with difference of distribution" (p. 786); see Z. Harris, *Distributional structure*, in *Papers in structural and transformational Linguistics*, Formal Linguistics Series, vol. 1, Humanities Press, New York, 1970, pp. 775-794.

class have a similar distributional behaviour, and that it was possible to analyse the whole of language according to the same criterion.

He agreed with Bloomfield regarding the impossibility, for a linguistic theory, to give a full account of meaning in all its social manifestations; but, even recognising the influence of extralinguistic factors on the use of language, Harris was also convinced that, in the measure in which a particular meaning is linguistic, it has a distributional correlate and it's therefore susceptible of analysis. [66]

The distributional properties of linguistic items, in Harris' view, were the answer to the question about the basis of semantic similarity: the more the contexts in which two words occur are similar, the higher is their semantic similarity. As in Bloomfield's work, there's the refusal of the meaning as *explanans* in linguistics: the similarities in the distributions of the linguistic elements, to the contrary, are the *explanans*, and they are the ground on which paradigmatic classes are built. If compared to the wide variety of semantic relations of traditional linguistic theory (synonymy, meronymy, antonymy, hyponymy etc.), the notion of semantic similarity could seem too general and not so explicative. But today, in cognitive science, this notion is more used than more specific ones, since it is graduable, unlike the traditional ones; futhermore there is a rich evidence of the effects of semantic similarity on the way we process the words that are stored in our mental lexicon. [67]

---

[66]My main references for history, issues and perspectives of distributional semantics are A. Lenci, *Distributional semantics in linguistic and cognitive research. A foreword*, in *Rivista di Linguistica*, vol. 20, no. 1, 2008, pp. 1-30; M. Sahlgren, *The distributional hypothesis,* in *Rivista di Linguistica*, vol. 20, no. 1, 2008, pp. 33-53.

[67]An example is the phenomenon called *semantic priming*, i.e. the time needed to the recognition of a target word is significantly less when another word, semantically similar, is presented to the subject just before the target.

One important question is concerning the nature of the relationship between the meaning of a word and its distribution: that is the principal point of distinction between the two versions of the Distributional Hypothesis.

- In the **weak Distributional Hypothesis**, the distributional analysis is a methodology for the study of the semantic paradigmatic properties of the linguistic items. Without doing any explicit assumption about the nature of meaning, in this weak version of the hypothesis we only suppose that the meaning of a word determines its distributional behaviour, i.e. there is only a *correlation* between meaning and linguistic distribution.

The weak DH is compatible with other theoretical frameworks of semantics, i.e. cognitive semantics and embodied cognition, since we have not to assume that "word distributions are themselves constitutive of the semantic properties of lexical items at a cognitive level" (Lenci: 2008). Indeed, in cognitive sciences there have been attempts of conciliation between embodied cognition and distributional semantics, consisting in mixed models in which both linguistic and senso-motorial information contribute to semantics. [68]

It is well-known that, in cognitive semantics, the conceptual representation of the world is intrinsically embodied, grounded in the sensory-motor systems; since concepts are modal entities, in this view, knowing the meaning of a word coincides with the ability to activate a

[68]Studies going in this direction are, for example, L. Barsalou, *Language and simulation in conceptual processing*, in *Symbols, embodiment and meaning*, edited by M. De Vega *et al.*, Oxford University Press, Oxford, 2008; G. Vigliocco *et al.*, *Toward a theory of semantic representation*, in *Language and* Cognition, vol. 1, no. 2, 2009, pp. 219-247.

simulation of our perceptual experiences of that entity [69].

Naturally, it is difficult to think that every concept in language is grounded in sensory modalities: abstract terms, like *democracy* for example, are not associated to perceptual experiences, nonetheless people know *how to use* them.

The hypothesis of a division of semantic labour between embodied cognition and distributional information could overcome the weaknesses of both the theories: on the one hand, the difficulties to explain those aspects of meaning that are not ascribable to senso-motorial experience; on the other hand, the inability to account for the aspects of the word meaning concerning reference to external world (it has been stressed by the opponents of the distributional semantics that meaning cannot be explained in terms of combination of symbols: it needs to be anchored to extralinguistic entities to which the words refer [70]). In models like Vigliocco's, semantic representations are generated from a statistical combination of experiential (in particular, sensory-motor) and linguistic information [71]. Thanks to this combination, "aspects of meaning learnt from linguistic data are not disembodied but become hooked up to the world" (Vigliocco *et alii*: 2009) , while it is presumable that distributional information assumes a central role in shaping other aspects of meaning that are not explainable through the reference to external reality.

- In the **strong Distributional Hypothesis**, there is a causal

[69]One of the most influential models for the theories of embodied cognition is the one proposed by Lawrence Barsalou, in L. Barsalou, *Perceptual symbol systems,* in *Behavioral and Brain Sciences*, no. 22, 1999, pp. 577-660.

[70]See, for example, A. Glenberg and M. Robertson, *Symbol grounding and meaning: a comparison of high-dimensional and embodied theories of meaning*, in *Journal of Memory and Language*, no. 43, Elsevier, 2000, pp. 379-401; A. Glenberg and S. Mehta, *Constraint on covariation: it's not meaning*, in *Rivista di Linguistica*, vol. 20, no. 1, 2008, pp. 237-262.

[71]G. Vigliocco *et al,* ibid., 2009.

relation between the distribution of a word and its meaning, in the sense that "repeated encounters with words in different linguistic contexts eventually lead to the formation of a contextual representation as an abstract characterization of the most significant contexts with which the word is used" (Lenci: 2008).

In this version, the hypothesis concerns the representation of word meaning on a cognitive level: in facts, the aim of some of the most influential models for distributional semantics, Latent Semantic Analysis (LSA) among the others (Landauer, Dumais: 1997), is to present a model -which is cognitively plausible- for the learning of word meaning through the extraction of regular co-occurrence patterns from the linguistic input [72]. In such a view, every encounter of a word contributes to its semantic representation, in the sense that it modifies its similarity relationships with the other words in our mental lexicon.

Obviously, the success of the DSMs in modeling the human ability to learn new words is strictly dependent on the dimension of the corpus and on the naturalness of the data: every word has to be related to tens of thousands of contexts, and the text selection has to reproduce, as much as possible, the natural linguistic input to which humans are exposed while they are learning a language.

2.2 The different fates of Distributional Semantics

[72] T. Landauer, S. Dumais, *A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge*, in *Psychological Review*, vol. 104, no. 2, 1997, pp. 211-240; see also T. Landauer, *On the computational basis of learning and cognition: Arguments from LSA*, in *The Psychology of Learning and Motivation*, edited by B.H. Ross, Elsevier, 2002, pp. 43-84; T. Landauer *et al.* (edited by), *The handbook of Latent Semantic Analysis*, Lawrence Erlbaum, Mahwah (New Jersey), 2007.

As widely discussed by Lenci in (Lenci: 2008), the distributional analysis of linguistic contexts has had very little fortune in theoretical semantics, since generativism, formal semantics and cognitive semantics have estabilished themselves as the most successful frameworks and have proposed totally different views on the formation of meanings and on the role of linguistic information.

On the one hand, generativism looks for the explanation of the linguistic structures in the cognitive principles governing the Universal Grammar, while the concrete usage of language is not considered as a reliable source of evidence (the emphasis is on the competence, and not on the performance [73]); on the other hand, as we have seen before, the concepts in cognitive semantics are intrinsically embodied, and the distributional properties of the linguistic items are not the *explanans* of the semantic representations, instead they are constrained by the mental processes through which we conceptualize the world.

Furthermore, there is another tradition whose approach to semantics is conflicting with distributional methods, i.e. the model-theoretic and referential semantics of authors like Frege, Tarski, Carnap,

[73]In the works of Noah Chomsky, the *competence* is the speaker-hearer's knowledge of his/her language, seen as a "mental reality" which is responsible for all the aspects of the language use which can be characterized as linguistic; competence manifests itself in the ability to produce and understand a theoretically infinite number of sentences , most of which the speaker-hearer has never heard before.

Instead, *performance* refers to the production of actual utterances. Only in an idealized situation the *performance* is a direct reflection of the *competence*, because it is affected by other factors like, for example, the nature and the limitations of the speaker-hearer's speech production and speech perception mechanisms, the nature and the limitations of his/her memory and concentration and other mental abilities, his/her social environment etc.

Linguistic performance is seen as "fairly degenerate in quality" (Chomsky: 1965), because the influence of all these factors makes it full of errors and deviations, and therefore it can't be considered as a reliable source of linguistic evidence.

See N. Chomsky, *Aspects of the theory of syntax,* MIT Press, Cambridge (MA), 1965.

Montague and many others. The criticism regarding the attempt of distributional semantics to examine the meaning and its properties in terms of combination of symbols, which are not hooked to referents in the external world, comes primarily from this side. In contrast, these models have proposed a denotational approach to semantics, that is conceived as the study of the "relation of signs to the objects to which the signs are applicable" (Morris: 1938) [74].

Within the corpus linguistics tradition, distributional semantics has had a completely different destiny: there was no need for a theoretical explanation of the adoption of the distributional hypothesis as a methodological principle; furthermore, to NLP researchers dealing with the problems related to lexical ambiguity, the distributional properties of the words are the only reliable criterion to discriminate between their possible senses. Adam Kilgarriff, for example, deals with a task having a long tradition in Natural Language Processing, Word Sense Disambiguation, which consists in disambiguating a word by choosing between a predefined set of senses given in a dictionary. Kilgarriff shows how every attempt to provide the concept of "word sense" with a solid theoretical foundation has failed, and proposes the construction of word senses as abstractions over clusters of word usage. [75]

Another reason for the success of distributional semantics in corpus linguistics was the *neo-empiricist turn* of the late 80s, which brought to the predominance of corpus-based statistical methods for language processing, also in tasks concerning lexical semantics.

[74]C. Morris, *Foundations of a theory of signs*, in *International Encyclopedia of Unified Science*, vol. 1, University of Chicago Press, Chicago, 1938, pp. 77-138.
[75]A. Kilgarriff, *ibid.*, 1997.

In the same period, the works of George Miller and Walter Charles in the psycholinguistic field, which presented one of the strongest assertions of the Distributional Hypothesis as a cognitive model of the form of semantic representations, helped the spreading of Distributional Semantics as well.

The cognitive representation of a word is some abstraction or generalization derived from the contexts that have been encountered. That is to say, a word's contextual representation is not itself a linguistic context, but is an abstract cognitive structure that accumulates from encounters with the word in various (linguistic) contexts.
The information that it contains characterizes a class of contexts (Miller, Charles: 1991). [76]

Unlike Harris, who had proposed distributional analysis as a scientific method for the study of meaning, but without further implications about the nature of the conceptual representations, Miller and Charles suppose that the origins of the representation of a word are to be sought in its repeated encounters in multiple linguistic contexts (therefore, we're speaking about a *contextual* representation). This representation includes all "*syntactic, semantic, pragmatic and stylistic conditions*" (Miller, Charles: 1991) governing

[76]G. A. Miller, W. G. Charles, *Contextual correlates of semantic similarity*, in *Language and Cognitive Processes*, vol. 6, no. 1, Taylor & Francis, 1991, pp. 1-28.

the use of that word [77]. In Miller and Charles' view, the semantic similarity between two words coincides with the degree of similarity of their contextual representation: the distributional properties of the words are not seen as a correlate of the meaning (whatever this might be); instead, they are considered as the semantic content itself.

2.3 From theory to praxis: Vector Space Models of semantics

Vector space models of semantics are a very popular framework in computational lexical semantics, and an ideal tool to implement the view of meaning as it is conceived in distributional semantics.

> The idea of Vector Space Models was proposed by Salton and colleagues in 1975, who designed an information retrieval system, named SMART, that anticipated many of the concepts used in the contemporary research on search engines. [78]

The task of document retrieval was conceived as one of word overlap between a query and a document, and the idea behind the Vector Space Model (from this point forward, VSM) is to represent each document in a collection as a vector in a vector space, where the basis vectors of the space are words; the query will be represented also as a vector, in the same space of the other documents (the query is a *pseudo-document*). The closer two vectors are, the higher the semantic similarity of the related documents will be; consequently, the documents in the collection are sorted in order of increasing

[77] On the same topic, see also W. G. Charles, *Contextual correlates of meaning,* in *Applied Psycholinguistics*, vol. 21, no. 4, Cambridge University Press, 2000, pp. 505-524.
[78] G. Salton *et al.*, *A vector space model for automatic indexing*, in *Communications of the ACM (Association for Computing Machinery)*, vol. 18, no. 11, 1975, pp. 613-620.

distance from the query and then presented to the user. [79]

The VSMs had a broad success in information retrieval, so that researchers tried to extend them to other semantic tasks in Natural Language Processing. In the most simple case, the first step in the creation of a VSM is the building of a frequency matrix, where each row corresponds to an event, each column corresponds to a particular context or situation and every cell of the matrix contains the number of times in which a particular event has occurred in that particular context / situation. Of course, the events in question are words, while the context / situation depends on the task.

The intuition in using the VSMs in Natural Language Processing was that two words have similar meanings when they tend to occur in similar contexts; since a whole document might not be the optimal length of a context for measuring word similarity, the context can be narrowed to a sentence, or to a small window of words including the target term.

Once we have shortened the context, the type of relation used to measure the similarity of the vector changes: while in a term-document matrix similar words will occur in the same documents, in a narrowed context it is difficult to think that similar words could co-occur; *vice versa*, since the writers try to avoid redundancies, it is difficult to think that near-synonymous words will appear in the same istance of a contextual window, because they will be used in alternance (Clark: 2012). Using a distinction that is stressed in (Sahlgren: 2008), we could say that the first approach -that is the

[79]See G. Salton, *ibid.*, 1975; P. Pantel, P. Turney, *ibid.*, 2010; S. Clark, *Vector Space Models of lexical meaning*, in *Handbook of Contemporary Semantics*, edited by S. Lappin, C. Fox, Wiley-Blackwell, 2nd edition, 2012.

typical one in information retrieval- uses *syntagmatic* relations between words to assess their semantic similarity. The assumption of the first approach is that words with a similar meaning will tend to occur in the same contextual unit, the document, because they are appropriate to define the particular topic of that document. If similar words are used in a query and in a document, this will probably mean that the document is relevant for the topic of the query.

Instead, the second approach uses *paradigmatic* relations, because in a small context window we don't expect that similar words (e.g. synonyms) can co-occur; but we could expect that their surrounding words will be -more or less- the same. [80]

The following is an example of a small corpus and of term-term matrix taken from (Clark: 2012): each sentence of the corpus is considered as a contextual window (the sentences are assumed to have been lemmatised during the matrix creation); the matrix cells contain the frequencies of co-occurence of the words.

An automobile is a wheeled motor vehicle used for transporting passengers.

A car is a form of transport, usually with four wheels and the capacity to carry around five passengers.

Transport for the London games is limited, with spectators strongly advised to avoid the use of cars.

The London 2012 soccer tournament began yesterday, with plenty

[80] According to the *refined distributional hypothesis*, formulated in M. Sahlgren, *ibid.*, 2008, *a distributional model accumulated from co-occurrence information containes syntagmatic relations between words, while a distributional model accumulated from information about shared neighbors contains paradigmatic relations between words*.

of goals in the opening matches.

Giggs scored the first goal of the football tournament at Wembley, North London.

Bellamy was largely a passenger in the football match, playing no part in either goal.

Term vocabulary: <*wheel, transport, passenger, tournament, London, goal, match*>

	wheel	*transport*	*passenger*	*tournament*	*London*	*goal*	*match*
car	1	1	1	0	0	0	0
automobile	1	2	1	0	1	0	0
soccer	0	0	0	1	1	1	1
football	0	0	1	1	1	2	1

Note that similar words (for instance *car* and *automobile*) tend not to co-occur within the same sentence, but they have highly similar vectors because they share the same neighbours.

Generally, the vector components are not the raw frequencies of words in contexts: words that are, in some way, "unexpected" should be weighed more than expected ones. In information theory, the surprising events are those with the highest information content [81]; in information retrieval, rare words are more informative than common ones and they are often useful to identify the topic of a document.

[81]C. Shannon, *A mathematical theory of communication*, in *Bell System Technical Journal*, University of Illinois Press, vol. 27, 1948, pp. 379-423, 623-656.

It is common, for these reasons, to replace the raw frequencies with some kind of weighting function.

Two of the most popular functions are:

- the **tf-idf** family of functions (term frequency * inverse document frequency), all based on the idea that a term has to get an high score when it is frequent in the corresponding document (i.e. the tf is high), but it is rare in other documents of the collection (i.e. idf is high). [82]

The tf-idf score is often combined with length normalization, because, if document length is ignored, the search engine will have a bias in favour of long documents. [83]

- the **Pointwise Mutual Information (PMI)**, which has proved to be a valid choice both for term-document and for word-context matrices.

Let *p(i)* be the probability of a word *i* in a corpus, and let *p(c)* be the probability of a context *c*; *p(i, c)* will be the joint probability of the two events, i.e. the probability that the word *i* occurs in the context *c*.

The PMI of the word *i* and the context *c* is defined as follows:

$$PMI(i,c) = \log \frac{p(i,c)}{p(i) * p(c)}$$

If the two events are statistically indipendent, *p(i, c) = p(i) * p(c)*, and consequently *PMI (i, c)* will be 0; but if there is an interesting semantic relation between the word *i* and the context *c*, they will

[82]G. Salton, C. Buckley, *Term-weighting approaches in automatic text retrieval*, in *Information Processing and Management*, vol. 24, no. 5, 1988, pp. 513-523.
[83]A. Singhal *et al.*, *Document length normalization*, in *Information processing and management*, vol. 32, no. 5, 1996, pp. 619-633.

occur more often than they would if they were indipendent, so that *p(i, c) > p(i) * p(c)*, and thus PMI will be positive. [84]

It should be evident, at this point, that the comparison between vectors is a key operation in the VSMs, because in this kind of model the distance in the space is the equivalent of the similarity of the meanings.

The most popular way to compare two frequency vectors (raw or weighted) is the cosine similarity. Let A and B be two vectors, each with *n* elements:

$A = <a_1, a_2, a_3 \ldots a_n>$

$B = <b_1, b_2, b_3 \ldots b_n>$

The cosine of the angle α between A and B is calculated as follows:

$$\cos(A, B) = \frac{\sum A_i * B_i}{\sqrt{\sum A_i^2 * \sum B_i^2}}$$

$$=\cos(A, B) = \frac{A*B}{\sqrt{A*A}*\sqrt{B*B}}$$

$$= \cos(A, B) = \frac{A*B}{|A|*|B|}$$

[84]K. Church, P. Hanks, *Word association norms, mutual information, and lexicography*, in *Proceedings of the 27th Annual Conference of the Association of Computational Linguistics*, Vancouver (British Columbia), 1989, pp. 76-83; P. D. Turney, *Mining the web for synonyms: PMI-IR versus LSA on TOEFL*, in *Proceedings of the 12th European Conference on Machine Learning (ECML 2001)*, Freiburg (GER), 2001, pp. 491-502.

The cosine of the angle between two vectors A and B, as we can see, is the inner product of the vectors after their normalization to unit length. Normalization is a necessary step, because one of the compared words could be very frequent and have a long vector, while the other could be rare and have for this reason a vector that is much shorter: comparing the cosines of the angle formed by the vectors allows us to measure the similarity of the words indipendently from vector length.

The value of the cosine ranges from -1, when the vectors are pointing in opposite directions, to 1, when they are pointing in the same directions; when the vectors are orthogonal, the cosine is 0. If we are dealing with raw frequency vectors, there will not be negative elements, and consequently the cosine will be positive; but negative elements can be introduced by weighting and smoothing operations.[85]

In addition to the cosine similarity, other similarity measures have been proposed: but it is commonly said in Information Retrieval that, once the vectors have been normalized in some way, the measure of distance makes no great difference. [86]

[85]See P. Pantel, P. Turney, *ibid.*, 2010, in particular pp. 156-162.

[86]See, for example, the Lin similarity presented in D. Lin, *Automatic retrieval and clustering for similar words,* in *Proceeding of the 17th International Conference on Computational Linguistics,* Association for Computational Linguistics, 1998, pp. 768-774.

2.4 Latent Semantic Analysis (LSA): a cognitive hypothesis

In order to improve information retrieval performance, we have to limit the number of vector components: computing the similarity between all pairs of vector is computationally too expensive, so we should compare only the vectors sharing a non-zero coordinate (vectors that do not share coordinates are obviously dissimilar). Very frequent grammatical words (such as *a*, *the*, *with*, *on*, *to*) will result matching a non-zero coordinate in most vectors, but their semantic relevance for identifying a particular topic is very low. Therefore, we need to use weighting functions like the PMI, so that we can assign high weights only to the dimensions representing highly discriminative contexts.

One of the most popular techniques to improve the performances of IR systems in calculating document similarity was proposed by Deerwester *et al.* in 1990, and was based on linear algebra: the truncated Singular Value Decomposition or SVD (Deerwester *et al.*: 1990). [87]

Let X be a term-document matrix, SVD decomposes it into the product of three matrices $U\sum V^T$, where U and V are in column orthonormal form (i.e. the columns are orthogonal and have unit length $U^T U=V^T V=I$) and $\sum$ is a diagonal matrix of singular values. If X is of rank *r*, $\sum$ will be of rank *r* too. 88

[87]S. Deerwester *et al.*, *Indexing by Latent Semantic Analysis*, in *Journal of the American Society for Information Science*, vol. 41, no. 6, 1990, pp. 391-407.

[88]The explanation is taken directly from P. Pantel, P. Turney, *ibid.*, 2010, pp. 158-160.

See also C. Manning *et al.*, I*ntroduction to Information Retrieval*, Cambridge University Press, Cambridge, 2008, pp. 406-417.

Let $\sum_k$, where $k < r$, be the diagonal matrix formed from the top k singular values, and let U_k and V_k be the matrices produced by the selection of the corresponding columns from U and V. It is demonstrated that the matrix $U_k \sum_k V_k$ is the matrix of rank r that best approximates the original matrix, i.e. it minimizes the approximation errors. 89

Deerwester and colleagues called *Latent Semantic Indexing* the application of this technique to document similarity: thanks to the dimensionality reduction, instead of a matrix with tens of thousands of documents and terms, they have to elaborate a low-rank approximation with only a few hundred basis vectors for each document.

As it was specified by the authors, truncated SVD could be applied not only to document similarity, but also to word similarity (Deerwester *et al.*: 1990).

The focus of the research of Deerwester and colleagues was on Information Retrieval tasks, so their work did not concern word similarity. To the contrary, Landauer and Dumais applied SVD to word similarity, presented their method -called *Latent Semantic Analysis*- as a cognitive hypothesis on human learning of word and passage meaning (consequently, they adopted the distributional hypothesis in its strong version) and provided the evidence that LSA could reach human-level scores on tasks like the multiple-choice synonym detection questions from the Test of English as a Foreign Language (TOEFL). 90

As claimed by the authors, LSA is seen as a solution to Plato's

[89]See G. Golub, C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.

[90]See T. Landauer, S. Dumais, *ibid.,* 1997; T. Landauer, *ibid.*, 2002.

problem (that is to say, the fact that we know much more than experience could have taught us), in the sense that *"It acquires linguistically and cognitively effective... representations of word meaning without any pre-existing specific linguistic knowledge"* (Landauer: 2002). It is presented as a plausible model of the acquisition, induction, and representation of linguistic knowledge: we learn language through the formation of contextual representations of the words, where the information about the most significant contexts of occurrence is included. Exactly like human beings, LSA must experience many contexts of occurrence of a word (like so many contexts in which the word does *not* occur) before learning its representation.

But how are the significant contexts identified? As we said in the first chapter, words seem to have prototypical contexts, which are probably the ones we think to when we assign them a polarity. So, the question could be: how are we presumed to select the contexts being more relevant to the representation of the word's meaning?

From this point of view, mapping the dimensions of the original matrix onto a lower-dimensional space is a fundamental operation, because "*very small dimensions (small singular values) represent very small, possibly locally unique components, larger ones the components that matter most in capturing similarities and differences*" (Landauer: 2002). In other words, the dimensionality reduction helps us to drop irrelevant dimensions (i.e. contexts of occurrence which are not relevant: for example, creative uses of the word in figurative speech are often too idiosyncratic to contribute to the contextual representation) and to preserve only those which contribute effectively to the construction of the word's meaning.

It is worth to point out that the top-*k* dimensions selected by SVD, in Landauer and Dumais' framework, do not correspond to the components of meaning of the componential semantics: they are the foundation upon which words are built, but they are not describable in words; we have to think of them as abstract features, not predictable on the basis of our intuition as speakers. Therefore, dropping dimensions that do not matter is important in order to preserve only a selected group of meaningful contexts, i. e. those contexts that are really relevant in the building of the meaning.

2.5 Different matrices for different tasks

We will finish this brief survey on VSMs by describing the different kinds of matrices.

As pointed out by Pantel and Turney, each type of matrix is particularly suited for specific tasks (Pantel, Turney: 2008). The authors mention main three types:

- **term-document matrices**, where the row vectors correspond to terms and the column vectors correspond to documents.

  As it is easy to imagine, this is the kind of matrix used in document retrieval to rank the documents in order of decreasing similarity between the query vector and the document vectors. The intuition is that documents having the same topic will probably use similar words.

  This kind of matrix is also used for such tasks as document

clustering 91, document classification (given a set of unlabeled documents and a training set of labeled ones - where the labels could correspond to topics, or sentiment orientations, or other kinds of classes-, the task is to learn from the training set how to assign labels to unclassified documents 92), automatical essay grading 93;

- **word-context matrices**, where the row vectors correspond to words and the column vectors correspond to their context.
- As noted by Deerwester *et al.*, in the vast majority of cases documents are not of the optimal length of text for the measurement of word similarity. Depending on the cases, the context can be given by words 94, grammatical dependencies 95 or more complex dependency links *96*; words that are highly similar will rarely co-occur, but they will tend to occur in similar contexts.

  As it is easy to understand, this kind of matrix is the most

[91]See, for example, D. Lin, P. Pantel, *Document clustering with committees,* in *Proceedings of the 25th Annual International ACM SIGIR Conference,* Tampere (Finland), 2002, pp. 199-206; G. Karpys, Y. Zhao, *Evaluation of hierachical clustering algorithms for documents datasets*, in *Proceedings of the 11th International Conference on Information and Knowledge Management,* McLean (Virginia), 2002.

[92]See, for example, F. Sebastiani, *Machine learning in automated text categorization*, in *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, 2002, pp. 1-47; S. Kim *et al.*, *Automatically assessing review helpfulness*, in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006, pp. 423-430.

[93]P. Foltz *et al.*, *The intelligent essay assessor: applications to educational technology*, in *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, vol. 1, no. 2, 1999.

[94]K. Lund, C. Burgess, *Producing high-dimensional semantic spaces from lexical co-occurrence*, in *Behavior Research Methods, Instruments and Computers*, vol. 28, no. 2, 1996, pp. 203-208.

[95]D. Lin, *Automatic retrieval and clustering of similar words,* in *Proceedings of the 17th International Conference on Computational Linguistics,* Association for Computational Linguistics, 1998, pp. 768-774.

[96]For a complete survey on this topic, see M. Sahlgren, *The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces,* PhD Thesis, Department of Linguistics, Stockholm University, 2006.

common for tasks connected to Word Sense Disambiguation; 97

- **pair-pattern matrices**, where the row vectors correspond to pairs of words, such as *on:off* and *dead:alive,* while the column vectors correspond to the patterns in which the pairs co-occur, such as "X or Y".

This kind of matrix was proposed by Lin and Pantel to measure the semantic similarity of patterns (Lin, Pantel: 2001): according to the *extended distributional hypothesis* they formulated, patterns co-occurring with similar pairs of words will tend to have similar meanings (their semantic similarity is measured as the cosine similarity of the corresponding column vectors). 98

Instead, Turney *et al.* used the pair-pattern matrix to measure the semantic similarity of relations between word pairs, i. e. the similarity of row vectors.

The authors formulated the so-called *latent relation hypothesis*, stating that pairs of words co-occurring in similar patterns tend to have similar semantic relations. 99

A relevant application of pair-pattern matrices is in tasks aiming at the classification of the semantic relations between the words; in a recent study, for example, Turney has showed the possibility to

[97]See, for example, T. Pedersen, *Unsupervised corpus-based methods for Word Sense Disambiguation,* in E. Agirre, P. Edmonds, *Word Sense Disambiguation: Algorithms and Applications*, Springer, 2006, pp. 133-166; D. Yuret, M. Yatbaz, *The noisy channel model for unsupervised word sense disambiguation*, in *Computational Linguistics,* vol. 36, no. 1, 2010, pp. 111-127.

[98]D. Lin, P. Pantel, *DIRT – Discovery of Inference Rules from Text,* in *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining,* San Francisco (California), 2001, pp. 322-328.

[99]P. D. Turney *et al.*, *Combining indipendent modules to solve multiple-choice synonym and analogy problems*, in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, Borovets (Bulgaria), 2003, pp. 482-489.

distinguish, through the use of a pair-pattern matrix, synonyms from antonyms, synonyms from non-synonyms *etc. 100*

It is useful to make a further distinction between the types of similarities: the term-document matrices and the word-context matrices measure the *attributional similarity* between terms/words, that is to say, they measure a degree of correspondance between the properties of terms/words; instead, the pair-pattern matrices measure the *relational similarity* between pairs of words *a:b* and *c:d*, which depends on the degree of correspondence between the relations of *a:b* and *c:d*. This kind of similarity is more relevant in the generation of resources such as the *thesauri* (i. e. WordNet), where the most of the information is given by the relations between the words, rather than in the individual words. 101

2.6 Unstructured and structured DSMs

Distributional Semantic Models, in their basic form, make use of two-way structures, i.e. matrices coupling target elements and contexts. According to Padò and Lapata, the core notion in the formal definition of a semantic space is a matrix $\mathbf{M}_{|B| \times |T|}$, where B is the set of the elements representing the contexts used to measure the

[100]See P. D. Turney, *A uniform approach to analogies, synonyms, antonyms and associations,* in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester (UK), 2008, pp. 905-912.
[101]See D. Gentner, *Structure-mapping: a theoretical framework for analogy*, in *Cognitive Science*, vol. 7, no. 2, 1983, pp. 155-170.

distributional similarity of the target elements T. 102

This kind of structure is typical of the approaches representing distributional data uniquely in terms of co-occurence relations between elements and contexts, known as **unstructured DSMs** because they register only the fact that a target element occurs in / close to a particular context, ignoring the type of the relation: the syntactic information is completely absent. For example, an unstructured VSMs analyzing the sentence *John saw a beautiful girl* would derive that *beautiful* and *girl* share the feature *see*, because they co-occur in the same context window; but, of course, there is not a linguistic relation between *see* and *beautiful*. 103

Instead, **structured DSMs** are syntax-aware: they extract data from corpora in the form of triples, generally two words linked by a lexico-syntactic pattern 104. The assumption is that the surface connection between two words is a cue of their semantic relation: consequently, the co-occurrence of two words is not enough to say something about the relation between their meanings, they have to be linked by some interesting patterns.

Note that, while in an unstructured DSMs there is a unique type of relation between words (the co-occurrence within the context window), different patterns correspond to different relations.

Typically, these models require a more refined corpus processing

---

[102]S. Padò, M. Lapata, *Dependency-based construction of Semantic Space Models,* in *Computational Linguistics*, vol. 33, no. 2, 2007, pp. 161-199.

[103]The distinction between structured and unstructured DSMs is discussed in M. Baroni and A. Lenci, *Distributional Memory: a general framework for corpus-based semantics,* in *Computational Linguistics,* vol. 36, n. 4, 2010, pp. 673-721.

[104]For example, Sketch Engine builds "Word Sketches", consisting of triples (word-word-relation) extracted from parsed corpora. Then, the number of shared triples is used to measure the attributional similarity between word pairs. See A. Kilgarriff *et al.*, *The Sketch Engine*, in *Proceedings of Euralex*, Lorient (FRA), 2004, pp. 105-116; Sketch Engine is available at http://sketchengine.co.uk.

(parsing, extraction of interesting patterns) and are more sparse, because there are more triples than pairs.

Even if structured DSMs extract a richer array of distributional information from corpora, it is still possible to represent it in the same way as unstructured DSMs, mapping the data onto a two-way matrix: for example, Pantel-Pennacchiotti (Pantel, Pennacchiotti: 2006) and Turney (Turney: 2006) concatenated the two words and used the links as contexts, in order to measure the relational similarity of the words 105; or the units formed by the concatenation of a word and a link can serve as a context to measure the attributional similarity between other words. 106

Another example of a structured DSM is the framework Distributional Memory, which will be presented more extensively in the next chapter: the distributional information is arranged in a third-order tensor, in the form of a weighted set of word-link-word tuples. DM is also the DSM we are going to use in this work to extract and analyze distributional information. 107

[105]Pantel and Pennacchiotti's and Turney's works can be seen as an application of the *latent relation hypothesis,* because the aim is to measure the relational similarity of word pairs through the analysis of the patterns in which they can occur, and words occurring in similar patterns will have similar semantic relations.

See P. Pantel, M. Pennacchiotti, *Espresso: leveraging generic patterns for automatically harvesting semantic relations,* in *Proceedings of COLING-ACL,* Sydney, 2006, pp. 113-120; P. D. Turney, *Similarity of semantic relations,* in *Computational Linguistics,* vol. 32, no. 3, 2006, pp. 379-416.

[106]Recent examples are A. Almuhareb, M. Poesio, *Attribute-based and value-based clustering: an evaluation,* in *Proceedings of the EMNLP*, Barcelona (SPA), 2004, pp. 158-165; K. Rothenhuäsler, H. Schütze, *Unsupervised classification with dependency based word spaces,* in *Proceedings of the EACL GEMS Workshop,* Athens (GRE), 2009, pp. 17-24.

[107]M. Baroni and A. Lenci, *ibid.*, 2010.

2.7 A distributional hypothesis for sentiment?

Until now, we have summarised the evolution of distributional semantics, from its origins rooted in Zellig Harris' methodological studies to the most recent progresses, represented by the vast number of cited studies implementing VSMs.

Now, we have to return to our starting point, i. e. the problem of determining the polarity of words and sentences. Curiously, in spite of the interest aroused by Sentiment Analysis, the use of distributional information for this purpose has been sporadic so far.

Yet the idea, that we could recognise a word's polarity by observing the polarity of the contexts in which it tends to occur, is quite intuitive. One of the few studies going in this direction is Turney and Littmann's, where two small sets of positive and negative words were selected, in order to measure the association strength of every target word with the words in the seed sets. The semantic orientation resulted from the sum of the scores of semantic association with the seed positive terms minus the sum of the scores of semantic association with the seed negative terms: if the sum is positive, the term is classified as having a positive polarity; otherwise, it is classified as having a negative polarity108.

The use of contextual information could help us to assign a polarity to a given word in, at least, two different ways:

- by exploring **syntagmatic relations**, i. e. how frequently that word co-occur with other words of a known polarity;
- by exploring **paradigmatic relations**, i. e. how similar are the

[108]P. Turney, M. Littmann, *ibid.*, 2003.

contexts of occurrence of that word and other words of a known polarity. If they share the same polarity, thy will tend to share the same neighbors.

In the continuation of this work, we are going to prefer paradigmatic relations in order to determine the semantic orientation of linguistic expression. In particular, we will try to build contextual vectors that can provide prototypical contexts of positive and negative words; then, we will use the similarity to these vectors as a cue of a positive / negative orientation.

Syntagmatic relations will not be ignored, but we will consider them in a further step and under a different perspective: we will explore the contextual modifications of the polarity, i.e. how words' semantic orientations combine themselves in contexts of close co-occurrence and in presence of syntactic bonds.

Other recent works have dealt with the problem of contextual polarity, and some of them have made similar assumptions.

For example, Maas *et al.* used a probabilistic model of documents which learns word representations, a model whose principles are similar to those of other probabilistic topic models, such as LDA [109]: the entries of every word vector were that word's association strength with respect to each latent topic dimension.

Since they used, as a training corpus, 25000 movie reviews from

[109]Latent Dirichlet Allocation is a popular probabilistic model in Natural Language Processing, explaining sets of observations in terms of latent variables. In the most common case, the observations are words collected into documents and the latent variables are the topics of the documents: the model assumes that each word's occurrence is explainable by reference to one of the document's topics.

This model was introduced in D. Blei *et al.*, *Latent Dirichlet Allocation*, in *Journal of Machine Learning Research,* vol. 3, no. 4, 2003, pp. 993-1022. See also: http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation.

IMDB, they built a predictor of sentiment polarity based on the star rating associated to every review on this site: if words tend to occur in documents whose average polarity is similar, they will have similar representations, and this can be seen as an application of the principle of the polarity of the contexts we mentioned before. 110

Also Das and Gambäck started from a similar hypothesis: all the words occurring in a certain syntactic territory tend to have the same semantic orientation. In their study, they aimed at the creation of a contextualized sentiment lexicon through a two-step process:

the first step is a network overlap technique, which finds overlaps of nodes between two lexical networks, ConceptNet and SentiWordNet 111. The algorithm starts with a SentiWordNet node (the concept corresponding to the word we have to disambiguate) and finds its closest neighbours in the ConceptNet network, considering the association strength between the node and its neighbours.

Then, Das and Gambäck used a SVM-based syntactic polarity classifier to assign contextual polarities to each association.

After the development of this semantic network, the contextual polarity of a sentence can be easily calculate as the sum of the association polarity scores of the concepts that are linked with dependency relations.

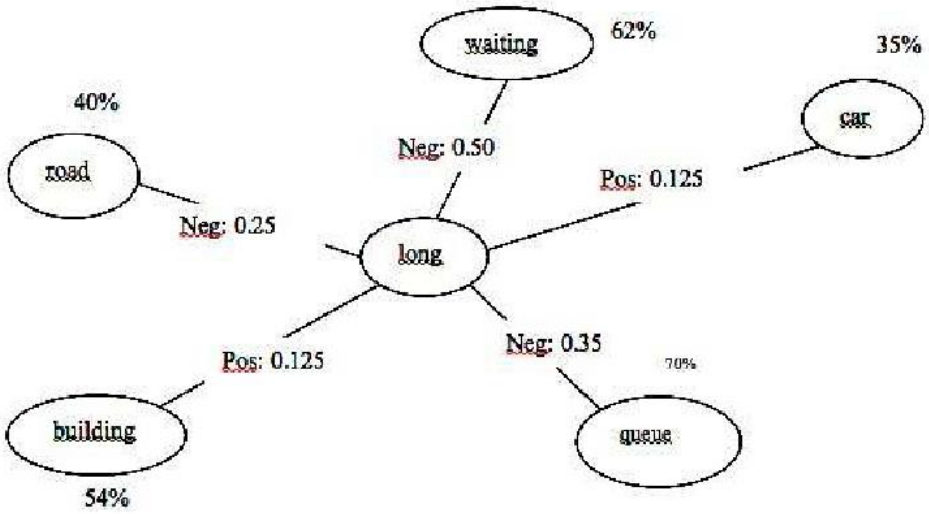
---

[110] A. Maas *et al.*, *Learning word vectors for Sentiment Analysis*, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, Stroudsburg (PA), 2011, pp. 142-150.

[111] ConceptNet is a "commonsense network", built from nodes representing concepts and labeled relationships between them. It was introduced by H. Liu, P. Singh, *ConceptNet: a practical commonsense reasoning toolkit*, in *BT Technology Journal*, vol. 22, no. 4, 2004, pp. 211-226.

This semantic network is available at http://conceptnet5.media.mit.edu/.

's approach



For example, the contextual polarity of the sentence

1. *I have been waiting in a long queue*

will be calculated as the sum of the association scores of the word *long* (the sentiment-bearing word, in this case) with *waiting* and *queue*. Since the scores are *Neg*: 0.5 and *Neg*: 0.35, the resulting contextual polarity will be *Neg*: 0.85;

In the second step, the researchers built a syntactic co-occurence network, using a clustering algorithm to partition a set of lexical entries into clusters of nodes, in order to extend their graph and to increase the coverage of the ambiguous sentiment terms. 112

In the following chapter, as we have anticipated, we will try to carry

[112]A. Das, B. Gambäck, *Sentimantics: Conceptual Spaces for Lexical Sentiment Polarity Representation with Contextuality,* in *Proceedings of the 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, Jeju (KOR), 2012, pp. 38-46.

out the classification of the polarity of sentiment words, leveraging paradigmatic relations: since words occurring in the same contexts tend to have the same semantic orientation, we could infer the polarity of a word from the polarity of its typical neighbours. Then, in the fourth chapter, we will deal with the problem of semantic compositionality and sentiment, analyzing how syntagmatic relations affect the semantic orientation of the components of a complex expression.[113]

[113]Even if it uses a totally different method, there is at least another work about contextual polarity in Sentiment Analysis deserving a mention: R. Socher *et al.*, *Semi-Supervised autoencoders for predicting Sentiment Distributions,* in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh (SCO), 2011, pp. 151-161.

Their approach is based on recursive autoencoders, neural networks able to learn a reduced dimensional representation of fixed-size inputs, that are used for sentence-level prediction of sentiment label distributions.

3
A distributional model to recognize the semantic orientation of single words

In their study of subjective meaning, Osgood and colleagues asked their subjects to rate words on a wide variety of scales, finding out that the semantic orientation of the words, i.e. their positivity or negativity, accounted for much of the variation in the data. [114]

Further studies on the semantic orientation (also known as *valence* in the linguistics literature) have highlighted that there is a high level of agreement among human judgements in the assignment of the labels: Hatzivassiloglou and McKeown labeled 1336 adjectives (657 positive and 679 negative words) and asked four people to indipendently label a sample of 500 of the adjectives of the testing set; on average, the subjects agreed that it was appropriate to assign a label to 89% of the 500 adjectives and, in such cases, they assigned the same label as the researchers to 97% of the terms. [115]

As we wrote in the first chapter, words probably have prototypical contexts and our judgements refer to words in those contexts. The agreement among the annotators could be interpreted as a proof that, for the majority of the words in the testing set, their most prototypical contexts have a clear semantic orientation.

---

[114] C. E. Osgood, G. Suci, P. Tannenbaum, *ibid.*

[115] V. Hatzivassiloglou, K. McKeown, *Predicting the semantic orientation of adjectives,* in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 18th Conference of the European Chapter of the ACL,* Association for Computational Linguistics, New Brunswick (NJ), 1997, pp. 174-181.

The cited article of Turney and Littmann presented a distributional semantic model: they use two different measures of word associations, the Pointwise Mutual Information (PMI), based on syntagmatic associations, and the Latent Semantic Analysis, based on paradigmatic associations and cosine similarity [116]. The assumption of the method is that the semantic orientation of a word is the same of its neighbours, and such an approach takes us back to Firth's maxim, which can be reformulated as follows: "You shall know *the semantic orientation of* a word from the company it keeps".

The approach we are going to follow is, in principle, similar to Turney and Littmann's LSA-based approach.

In the following pages, we will describe an experiment of polarity assignment performed on a set of previously labeled terms.

3.1 Data preparation

For the experiment, we needed a list of words labeled with their respective part-of-speech and semantic orientation.

We used the Opinion Finder's subjectivity lexicon, a list of subjective clues which is part of the Opinion Finder and is available for download [117]. These clues were extracted from several sources; some of them were manually compiled, while others were identified automatically using both annotated and unannotated data: examples are the part of speech, the prior polarity of the word, the strength of

[116] P. Turney, M. Littman, *ibid.*

[117] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

the subjectivity expressed by the word etc. [118]

Through the use of regular expressions in *GNU emacs* [119], We removed all the unnecessary information and processed the input file in order to obtain, for every word, a format of the kind *word PoS Polarity* (the value of this column is -1 for negative words, 1 for positive words):

```
...
abandoned    adj     -1
abandonment          noun   -1
abandon      verb    -1
abase  verb    -1
abidance     noun    1
...
```

The original file contained a list of 8221 positive and negative words belonging to four different parts of speech: noun, adjectives, verbs and anypos, a general class including words that can belong to more than a single PoS.

Then, we removed all the words labeled as anypos (1147 words), because we decided to use only the words having a single PoS; we also removed the words whose prior polarity was labeled as "neutral" or "both" (595 words).

After having "cleaned" my file, we wanted to filter my words by frequency. On the Corpus Query System Sketch Engine, it is possible to count the occurrences of words through the feature "Word List": the interface allows to specify a "white list" of words (contained in a file uploaded by the user), whose number of

[118] T. Wilson, J. Wiebe, P. Hoffmann , *ibid.*.

[119] See http://www.gnu.org/software/emacs/,

occurrences has to be counted.

We chose the corpus UkWac, and we selected as a lower bound the number of 1000 occurrences. In the resulting list, there were1403 positive words and 1455 negative words left, for a total of 2858. [120]

In order to measure the semantic orientation of the words, we used Distributional Memory, a framework for the extraction of distributional information from a large corpus in the form of a set of weighted of word-link-word tuples arranged in a third-order tensor. From the tensor, different matrices can be generated, in order to deal with different semantic problems by using always the same source of semantic information. [121]

The authors made the full Type Distributional Memory labeled tensor (30686 words) available, together with a *word-by-link-word* matrix in a version compressed by Random Indexing (the words are represented by 5000 dimensional vectors approximating the vectors of higher dimensionality of the tensor), that can be used for efficient computation of word-to-word (attributional) similarity, and with the top-10 nearest neighbours of each word from the random indexed matrix [122]. They also provided a set of Perl scripts to manipulate the data extracted from the tensor: there is the possibility to build

[120] See http://www.sketchengine.co.uk/.

[121] For a complete description of the Distributional Memory framework, see M. Baroni and A. Lenci, *ibid.* 2010.

[122] *Random indexing* is a dimensionality reduction technique, based on the insight that high-dimensional Vector Space Models implementations are unpractical because of their high computational cost, and that a high-dimensional model can be projected into a space of lower dimensionality without relevant modifications in the distances, if the resulting dimensions are chosen appropriately.
See M. Sahlgren, *An introduction to Random Indexing*, in *Proceedings of the Methods and Applications of Semantic Indexing Workshop,* at the *9th International Conference on Terminology and Knowledge Engineering*, Copenaghen (DEN), 2005.

matrices from the tuples, to sum vectors, to calculate the similarity of words *etc.* [123]

In the tensor we found the data that we needed to build a distributional "sentiment" space.

One of the downloadable scripts, *filter_by_field.pl*, takes as input files:

- a target list of words in a one-string line format;
- another list, where each line must have the same number of fields, and return an output file where the only lines left are those that in the second file had the element in the field *n* identical to one of the strings in the target list. [124]

The script filtered out some words: out of the 2858 words of the input file, only 1813 (838 positive, 975 negative) were present in the tensor.

At this point, we had two lists of positive and negative words represented as vectors in a distributional space. Their semantic orientation is known, because they have been classified by the researchers who assembled the sentiment lexicon, so they could serve us as a test set for our attempts of polarity classification.

[123] See http://clic.cimec.unitn.it/dm/.

[124] All the scripts we used will be described in detail in the appendix.

3.2 Related work

In their work Turney and Littmann, after selecting a positive and a negative seed set, each one of seven words, measured the semantic association of a target word with every word of the seed set [125]. Then, they calculated the semantic orientation as follows:

$$SO = \sum Association(targetword, positivewords) - \sum Association(targetword, negativewords)$$

The target word was classified as being positive if SO > 0, and it was classified as being negative otherwise.
The words of the seed sets, chosen by the authors because of their lack of sensitivity to context, were the following ones:

Pwords = {good, nice, excellent, positive, fortunate, correct, superior}
Nwords = {bad, nasty, poor, negative, unfortunate, wrong, inferior}

Since these words have the same orientation in almost all contexts, they can be considered as prototypical words for their respective polarity. Note that all the terms selected by Turney and Littmann are adjectives: if we look for "polarized" words, it is probably more natural to think of adjectives, because they are commonly used to evaluate (positively and negatively) entities and actions. This is

[125] P. Turney, M. Littman, *ibid.*

perfectly coherent with some important assumptions of the most part of the approaches to Sentiment Analysis: the adjectives are considered to be the best indicators of the subjective content of a sentence, and the semantic orientation of sentences and documents is often calculated as a linear combination of the "polarity scores" of all the adjectives that are present.

The researchers used two different measures of semantic association, the Pointwise Mutual Information and the Latent Semantic Analysis.

In the first case, they have estimated PMI by issuing queries to the search engine AltaVista and noting the number of the hits (matching documents). AltaVista was chosen over other search engines because of its NEAR operator, which allowed to search only the documents where the words co-occurred within a window of ten words, in either order.

In the second case, they used an online demonstration of LSA [126] and they chose the TASA-ALL, which was the largest among the available corpora (it is a set of short English documents gathered from a variety of sources by Touchstone Applied Science Associates, and it contains approximately 10 million words); starting from the corpus, a word-context matrix was generated and then SVD was used to reduce it to 300 dimensions.

In their experiment with the two measures, Turney and Littmann used two different lexicons and three different corpora: the performance of the PMI and of the LSA were compared on the corpus TASA with the lexicon of Hatzivassiloglou and McKeown.

[126] Available at http://lsa.colorado.edu/.

Size of test set	Accuracy of PMI	Accuracy of LSA
1336 words	61,80%	67,66%
1002	64,17%	73,65%
668	46,56%	79,34%
334	70,96%	88,92%

As we can see, the LSA outperforms PMI indipendently of the dimension of the test set.

To resume briefly the results of the comparison:

- the performances are near when evaluated on the full test set, but the LSA is remarkably better when the percentage of the test set is decreased;
- PMI seems to be less stable than LSA, especially when the percentage drops below 75%.

3.3 Building the prototypes

Similarly to Turney and Littmann's work, we chose sets of words that can be considered prototypical of a positive or negative semantic orientation. But, instead of calculating the scores of semantic association (PMI) or of semantic similarity (LSA) of a target word with each of the terms of the seed set, we used the Distributional Memory framework to extract the vectors of the seed words; then with the script *sum_vectors.pl,* taking as input a list of couples of the type vector ID-set ID (in our case, the sets were POSITIVE_SEEDS and NEGATIVE_SEEDS) and a matrix (our

distributional sentiment space), we built the centroid of the vectors of the words of the seed sets, which we will consider as the protoype vectors of the two poles of positivity and negativity.
Turney and Littmann classified the words by measuring the sum of the association / similarity scores of every target word *t* with the words of the seed set: if the sum of the scores for the positive set was higher, the target word *t* was considered to be positive; otherwise, it was considered to be negative. Similarly, we classify every target word *t* by assigning to it the label of the pole whose prototype vector is nearer to the vector *t*.
In order to compare the results, we will also make an attempt with Turney and Littmann's measure: instead of calculating a centroid for a seed set of vectors, we will calculate the cosine similarity of the vectors of the targets with the vectors of each seed word, and we will sum the scores; if a target word vector has an higher overall similarity with positive vectors, it will be classified as positive; otherwise, it will be classified as negative.

For this classification task, we used the 1813 words of the test set (838 positive and 975 negative), excluding from time to time those that were derived from some words in the seed set. We measured for every test word vector the distance from the positive prototype and from the negative and we assigned the label of the prototype with the highest cosine similarity; then, we compared the labels with those manually assigned by the creators of the lexicon. The cosine similarity has been calculated through the script *compute_cosine_of_pairs.pl*, which returns the similarity scores of a list of couples of vectors indicated through their IDs.

Finally, we wrote a simple script in Python, taking as input the original lists (compiled by the authors) of positive and negative words and their respective cosine similarities with the two prototypes, which calculates the Accuracy as the number of correctly classified words over the total.

3.4 The seed sets

For my experiment, we used two different seed sets.
The first one is the same used by Turney and Littmann:

Pwords = {good, nice, excellent, positive, fortunate, correct, superior}
Nwords = {bad, nasty, poor, negative, unfortunate, wrong, inferior}

As stated by the authors, these words were selected "*for their lack of sensitivity to context*", in the sense that they are positive or negative in almost all contexts; furthermore, the sets consists of opposing pairs (*good/bad*, *positive/negative etc.*). [127]
Note that the terms chosen by the researchers are all adjectives, and their sense is very generic (i.e. they can be used in a great variety of contexts).
We selected the second seed set from the NRC word-emotion

[127] P. Turney, M. Littman, *ibid.*

association lexicon, created by Saif and Turney [128]. This lexicon has human annotations of emotion associations for more than 14200 word types, and the annotations include whether the target is considered to be positive or negative, and whether it has associations with eight basic emotions (*joy, sadness, anger, fear, surprise, anticipation, trust, disgust*).

In order to identify the words for our seed set, we followed this procedure: first, we divided the words associated to a positive target and/or to positive emotions and the words associated to negative ones in two different files (excluding *surprise* and *anticipation,* that express kinds of "neutral emotions", *joy* and *trust* were considered as positive emotions, while *sadness, anger, fear* and *disgust* as negative ones); then we calculated the positive / negative score for each word by adding one point for every association with a positive / negative emotion; finally, among the terms with the highest scores for one of the two polarities, we chose the words of my seed sets.

The resulting words belonged to different parts-of-speech. Therefore, instead of using only two seed sets, we chose to create six of them: positive and negative nouns, positive and negative adjectives, positive and negative verbs.

The vectors of every word in the test set were compared only to the prototype vectors of their same part-of-speech.

[128]The lexicon was introduced in: S. Mohammad, P. D. Turney, *ibid*. The articles describing the genesis of the resource are available at: http://www.umiacs.umd.edu/~saif/WebPages/ResearchInterests.html, while a copy of the lexicon should be requested directly to Dr. Muhammad Saif (saif.mohammad@nrc-cnrc.gc.ca).

Positive seed sets

PositiveVerbs = {achieve, enjoy, encourage, share, improve, pay, save}
PositiveNouns = {friend, church, income, hope, respect, mother, money}
PositiveAdjectives = {perfect, excellent, safe, kind, pretty, happy, true}

Negative seed sets

***NegativeVerbs* = {sin, lose, hate, threaten, murder, abuse, slaughter}**
NegativeNouns = {hell, death, disaster, discrimination, poverty, cancer, terrorism}
NegativeAdjectives = {terrible, mad, bad, ill, illegal, adverse, hanging}

3.5 Results and observations

The results of the experiments can be observed in the following tables:

Size of test set	**Accuracy of PMI**	**Accuracy of LSA**
1336 words	61,80%	67,66%
1002	64,17%	73,65%
668	46,56%	79,34%
334	70,96%	88,92%

Positive words	**Total number of words**	**Correctly classified**	**Accuracy**
Adjectives	441	340	77,09%
Substantives	273	182	66,66%
Verbs	121	67	55,37%
Total (all the PoSP)	835	589	70,53%
Negative words	**Total number of words**	**Correctly classified**	**Accuracy**
Adjectives	431	336	77,95%
Substantives	368	279	75,81%
Verbs	172	115	66,86%
Total (all the PoSN)	971	730	75,18%
Totals	1806	1319	73,03%
Positive words (RI)	**Total number of words**	**Correctly classified**	**Accuracy**
Total (all the PoSP)	835	416	49,82%
Negative words (RI)	**Total number of words**	**Correctly classified**	**Accuracy**
Total (all the PoSN)	971	492	50,67%
Totals	1806	908	50,27%

Table 1: Results (Turney-Littmann's seed set)

Positive words	**Total number of words**	**Correctly classified**	**Accuracy**
Adjectives	441	331	75,05%
Substantives	271	228	84,13%
Verbs	121	36	29,75%
Total (all the PoSP)	833	587	71,42%
Negative words	**Total number of words**	**Correctly classified**	**Accuracy**
Adjectives	431	271	62,87%
Substantives	368	254	69,02%
Verbs	168	158	94,04%
Total (all the PoSN)	967	683	70,63%
Totals	1800	1270	70,55%

Table 2: Results (Seed set derived from Saif-Turney's emotion lexicon)

Positive words	**Total number of words**	**Correctly classified**	**Accuracy**
Adjectives	440	400	90,90%
Substantives	272	261	95,95%
Verbs	121	92	76,03%
Total (all the PoSP)	833	753	90,39%
Negative words	**Total number of words**	**Correctly classified**	**Accuracy**
Adjectives	431	233	54,06%
Substantives	368	151	41,03%
Verbs	171	46	26,90%
Total (all the PoSN)	970	430	44,33%
Totals	1803	1183	65,61%

Table 3: Results (Turney-Littmann's seed set, Turney-Littmann's original method)

The results obtained with the two seed sets were very different: our "classifier" had quite good performances with the seed set of Turney and Littmann, and these performances were stable indipendently from the part-of-speech. Remember that the results of Turney and Littmann, for a test set of comparable size (1336 words), were respectively 61,8% (PMI-based accuracy) and 67,66% (LSA-based accuracy). Moreover, even if our method -like LSA- was based on vector spaces, we did not use the Singular Value Decomposition, which could further improve these results. We tried a different dimensionality reduction method, Random Indexing, hoping to smooth the data and to improve our results, but our attempt was unsuccessful [129]: with a space reduced to 5000 dimensions, the global performance of the classifier was similar but slightly worse, with an increase of the accuracy of classification for positive words and a decrease for negative words. [130]

[129] See M. Sahlgren, *ibid.*, 2005.

[130] We also tried with 10000 dimensions, but the performance of the classifier was really poor.

Negative words	Score	Positive words	Score
disastrous-j	0,181	superb-j	0,181
horrendous-j	0,179	popular-j	0,153
horrific-j	0,177	fine-j	0,153
dreadful-j	0,175	modern-j	0,148
severe-j	0,172	stylish-j	0,148
appalling-j	0,170	special-j	0,146
dire-j	0,168	original-j	0,144
fatal-j	0,164	elegant-j	0,143
distressing-j	0,155	suitable-j	0,136
damaging-j	0,144	classic-j	0,134

Table 4: top-10 negative and top-10 positive words (Turney-Littmann's seed set)

The lack of stability seems to be the main fact, if we look at the results obtained with the second seed set. This is particularly evident in the verb classification: almost all the negative verbs were correctly classified, while the Accuracy for the positive verbs drops to less than 30%. Initially, we thought that the problem, with the positive verbs, was the choice of the words of the seed set. But even after many attemps with different seed sets, the performance did not improve significantly. This is not a totally surprising result: verbs are probably the parts of the speech that are most sensitive to the modulation of the context, at least with regard to semantic orientation. The performance of Turney-Littmann's seed set are better, but the verbs are still the most difficult PoS to classify.

For the adjectives, the performances of our "classifiers" are generally good (in three cases over four, the Accuracy is over 70%)

and a seed set made only of adjectives, as Turney and Littmann's, seems to work well even for the polarity classification of words belonging to a different PoS. This is coherent with the intuition that the adjectives carry the most of the subjective content of a sentence: if the sentiment can be studied on distributional basis, words' syntagmatic associations with positive and negative adjectives are probably one of the most precious sources of information for determining their polarity.

We have also made an attempt with Turney and Littmann's measure, considering the similarities with the single seed vectors, and not the similarity with a "polarity prototype".

The main feature of results, shown in Table 3, is again the lack of stability: while the performance is very good for the classification of positive words (the precision for adjectives and substantives is over 90%), the method fails to classify the negative ones, with percentages below 50% both for the overall precision and for the single PoS.

Again, the verbs seem to be the most context-sensitive part of speech, and consequently the most difficult to classify: the performance of the classifier is the worst for both positive and negative words; in the second case, the precision drops even below 30%.

Why does this happen? First of all, we have to remember that we are leveraging the paradigmatic similarity of the words, and not the frequency of their syntagmatic association: in spite of having the same polarity, two words can still have a different distributional behaviour, and this is reflected by a low similarity score. Note that the words of Turney and Littmann's seed set are all adjectives, and

that the performances on the adjectives are generally good.

The sentiment classification through the comparison with a prototype, built as the centroid of the vectors of a seed set, has a better performance compared to the same task carried out through the comparison with the single vectors. Maybe, the reason could be that the prototype vector has an higher number of non-zero entries (remember that the cosine similarity of two vectors is calculated as their dot product over the product of their magnitudes: if an entry in one of the two vectors is 0, the contribution of that dimension to the overall similarity will be 0 too) and, perhaps, these extra non-zero entries correspond to semantically-oriented contexts, that give a relevant contribution to the overall similarity score.

Finally, we extracted some top lists of the most positive and negative words, according to the results of each of the tested approaches (the lists are reported in the appendix).

At the end of this chapter and after having seen the results, some considerations can be done also on the method: the approaches used in (Turney, Littmann: 2003) are respectively based on syntagmatic and on paradigmatic associations, while our approach only takes advantage of the paradigmatic similarity (that means, in our context, the tendency to share neighbours of a certain polarity) between words having the same semantic orientation. [131]

A seed set made of generic words, with very high frequencies, seems to perform better than seed sets made of specific words, i.e. occurring in a more limited range of contexts, even if their positivity or negativity is highly marked (for example, words such as *cancer*

[131]Again, for the distinction among the two types of models, see M. Sahlgren, *ibid.*, 2008.

and *terrorism* are probably more negative than any word of Turney and Littmann's seed set; nevertheless, the classification method based on the last one works better).

Furthermore, a prototype built from a seed set of words is more effective for such a classification task than the seed words, taken in isolation: a possible explanation of this fact is that the prototype's vector, being the result of the sum of more vectors, presents a wider range of non-zero entries corresponding to semantically-oriented contexts, providing thus a better "basis" on which we can carry out our sentiment classification task.

4
Sentiment Analysis and Compositional Distributional Semantics

In language, words combine together to form more complex linguistic expressions.

Compositionality is considered as a basic principle governing the interpretation of these complex expressions, and it can be formulated as follows:

1. *The Principle of Compositionality*

The meaning of a complex expression is a function of the meanings of its parts of the syntactic rules by which they are combined. (Partee *et al.*: 1990, p. 318) [132]

This principle derives from two fundamental presuppositions:

a) a language has an infinite number of grammatical sentences;
b) a language has unlimited expressive power.

It is evident that an infinite number of sentences cannot be stored -as asort of "sentence dictionary"- in a finite brain, and consequently this number must arise from the combination of a finite list of elements, according to generative rules (at least some of which are recursive). If the meanings of the sentences were not composed in rule-governed ways out

[132]B. H. Partee, A. Ter Meulen, R. E. Wall, *Mathematical methods in linguistics*, Kluwer, Dordrecht (NL), 1990.

of meanings of their parts, they probably would not be interpretable. [133]

Vector-based models are generally directed at representing words in isolation, and the representation of more complex linguistic units is a topic that has received, until now, relatively little attention.

Previous work in this field has focused on the analysis of the operations used to approximate the composition of word meanings: in particular, the so-called *compositional distributional semantics models* [134] try to obtain distributional meaning for sequences through some kind of composition of the vectors of the single words in the sequence.

Of course, such an operation has a lot of parameters that cannot be easily estimated.

The ability of modeling the transformation of meanings due to semantic composition is fundamental for the aims of sentiment analysis, since we do not meet words in isolation, but more complex linguistic structures in which the prior polarities of the words are modified by the action of the context. In particular, we are interested in those phenomena of meaning composition causing polarity shifts, i.e. modifications of the prior polarities of the words.

[133] A. Cruse, *ibid.,* 2004; see, in particular, chapter 4.

[134] To my knowledge, the expression was introduced in F. M. Zanzotto *et al.*, *Estimating linear models for Compositional Distributional Semantics*, in *Proceedings of the 23rd International Conference on Computational Linguistics,* Association for Computational Linguistics, Bejing (CHN), 2010, pp. 1263-1271.

4.1 Compositionality in distributional semantics: the state of the art

The problem of the composition of word vectors received some attention in the connectionist literature, especially in response to criticisms of the ability of connectionist representations to handle complex structures: neural networks can easily represent single distinct objects, but in case of multiple objects they keep track with difficulty of which features are tied to which objects. This binding problem grows worse in the field of natural language processing, because of the hierarchical structure of language: as pointed out by Fodor and Pylyshyn, for a connectionist approach it would not be easy to distinguish between sentences such as *Mary loves John* and *John loves Mary*, because they have the same participants but different structures (the network of nodes would fail to keep track either of the fact that the same participant has a different role in the two sentences, or of the fact that the sentences involve the same participants, because e.g. Mary as a subject would have a distinct representation from Mary as an object*)*. [135]

On the contrary, symbolic models are able to handle the binding of the constituents to their roles in a systematic manner, and consequently to represent complex, hierarchical structures. [136]

Schematically, the vector-based approaches to semantic

[135]J. Fodor, Z. Pylyshyn, *Connectionism and cognitive architecture: a critical analysis,* in *Cognition*, vol. 28, Elsevier, 1988, pp. 3-71.

[136]For an overview of this debate, see J. Mitchell, M. Lapata, *Vector-based models of semantic composition*, in *Proceedings of the Association of Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Columbus (Ohio), 2008, pp. 236-244; J. Mitchell, M. Lapata, *Composition in distributional models of semantics,* in *Cognitive Science,* vol. 34, no. 8, 2010, pp. 1388-1439.

compositionality can be classified on the basis of the kind of operation used to compose the vectors. Consequently, we can identify "families" of models such as:

- additive models;
- multiplicative models;
- tensor product-based models;
- regression-based models;
- kernel-based models. [137]

In the literature on Information Retrieval, vector addition is the most popular method for the representation of the composed meaning of linguistic sequences, which is modeled as the sum of the single word vectors. 138

In the vector addition models, given two indipendent vectors *v1* and *v2,* their compositional meaning *v3* consists of the sum of the corresponding components of the original vectors:

$$1.v3_i = v1_i + v2_i$$

The vector addition does not increase the dimensionality of the resulting vector, but it is order indipendent, so it fails to capture differences of meaning due to the syntactic structure. To deal with this problem, alternative models for vector addition have been proposed, for example, by Kintsch: the basic idea of his work is to

[137]Complete reports on the different approaches to vector-based compositional semantics can be found in: J. Mitchell, M. Lapata, *ibid.*, 2008; K. Erk, S. Padó, *ibid.*, 2008; J. Mitchell, M. Lapata, *ibid.,* 2010; E. Guevara, *Computing semantic compositionality in distributional semantics,* in *Proceedings of the 9th International Conference on Computational Semantics,* Association for Computational Linguistics, Stroudsburg (Pennsylvania), 2011, pp. 135-144.

[138]See D. Widdows, *Geometry and Meaning,* CSLI Publications, Stanford, 2004, in particular chapter 5.

model the way the meaning of a predicate changes depending on the arguments it operates upon; to reach this aim, he suggested to add not only the vectors representing the predicate and its arguments, but also the neighbours associated to them, in order to "strengthen the features of the predicate that are appropriate for the argument of the predication" (Kintsch: 2001). 139

By contrast, in the models based on vector pointwise multiplication, each corresponding pair of components of the vectors *v1* and *v2* is multiplied to obtain the corresponding component of the resulting vector *v3*:

$$2.v3_i = v1_i * v2_i$$

Multiplicative models, as far as I know, are rarely implemented in this basic form: while additive models consider all the available components to capture the compositionality of the meanings, a simple multiplicative model only consider the non-zero entries of the original vectors. But, in their experiments, Mitchell and Lapata used both a simple multiplicative model and a weighted combination of the additive and the multiplicative model (*x*, *y* and *z* are weighting parameters):

$$3.v3_i = xv1_i + yv2_i + z(v1_i * v2_i)$$

their aim, with this combined model, was to avoid one of the

[139]W. Kintsch, *Predication,* in *Cognitive Science,* vol. 25, no. 2, 2001, pp. 173-202.

Note that the aim of Kintsch's work is not to create a general vector space model for semantic compositionality, but only for predicate-argument combinations.

potential drawbacks of pointwise vector multiplication , i.e. the effect of components with value zero. Surprisingly, their results in a similarity task proved that the two models perform equally well.
Futhermore, Mitchell and Lapata defined in their work a general class of vector-based models for semantic composition on the basis of four parameters:

$$4.p = f(u, v, R, K)$$

- *u* and *v* are the constituents of a complex expression;
- *p* is the complex expression whose constituents are *u* and *v*;
- *f* is the function used to model the semantic composition of *u* and *v*;
- R is the syntactic relation linking *u* and *v*;
- K is the the additional knowledge that is needed to construct the semantics of the combination of the components.

The authors noticed that, in order to limit the number of the functions that we have to consider, this framework can be simplified: most of the studies have focused only on a specific syntactic structure (most frequently, verb-object or adjective-noun), and in such cases R is hold fixed; and K can be ignored, because in most of the related work compositionality is modeled only on the basis of the semantic content of the components, without any additional knowledge. Mitchell and Lapata took in consideration mostly additive and multiplicative models (and their combination): in their case, the space of the considered functions can be narrowed

further, assuming *f* to be linear. [140]

Another trend of studies considers the tensor product as the basic operation to model compositionality: Smolensky was the first to propose it as a method to bind one vector to another: the tensor product is matrix *v1 x v2* is a matrix whose components are all the possible products $u_i v_j$ of the entries of vectors *v1* and *v2*. 141
However, this method produces a matrix whose dimensionality is higher than the dimensionality of the original vectors (and the dimensionality rises with every word added to the representation) 142. Therefore, further studies have proposed a method, based on circular convolution 143, to compress the tensor product of the word

[140]J. Mitchell, M. Lapata, *ibid.*, 2008; J. Mitchell, M. Lapata, *ibid.,* 2010.

Note that, in these works, Mitchell and Lapata deal mainly with additive and multiplicative models, but the general formulation of the framework allows to construct the composition of the vectors in a distinct space from *u* and *v*; consequently, even tensor product-based models can be included in the general class described above.

It is worth mentioning that, in more recent works, a compositional matrix-space model of language, using matrices instead of vectors and matrix multiplication as the unique composition operation, has been proposed by Rudolph and Giesbrecht.

See, for example, E. Giesbrecht, *Towards a matrix-based distributional model of meaning,* in *Proceedings of Human Language Technologies: the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics – Student Research Workshop,* Association for Computational Linguistics, 2010; S. Rudolph, E. Giesbrecht, *Compositional matrix-space models of language,* in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Uppsala (SWE), 2010, pp. 907-916.

[141]P. Smolensky, *Tensor product variable binding and the representation of symbolic structures in connectionist systems*, in *Journal of Artificial Intelligence,* vol. 46, 1990, pp. 159-216.

[142]Given two vectors **U** and **V**, their tensor product **U x V** is a matrix whose ij-th entry is equal to $\mathbf{U_i}$ **x** $\mathbf{V_j}$: see http://en.wikipedia.org/wiki/Tensor_product.

[143]The circular convolution is a mathematical operation able to compress the tensor product of two vectors onto the original space.

Given two vectors **U** and **V**, their circular convolution **U x V** is:

$$Z = \sum_{j=0}^{i-1} u_j v_{i-j}$$

See http://en.wikipedia.org/wiki/Circular_convolution.

vectors onto the original space. 144

As pointed out by Guevara, a point in common of the approaches outlined above lies in the fact that all the semantic content of the composed expression *v3* results from the combination of the single word vectors *v1* and *v2* (but sometimes the meanings of the components are not sufficient to account for the meaning of a complex linguistic expression, and some extra knowledge is needed: think, for example, to an idiomatic expression like *not to have the stomach*) 145; furthermore, all these approaches rely on the application of a single geometric operation on the components, and it is difficult to think that just a geometric operation could account for all the possible semantic transformations due to compositionality. 146

Some recent works have tried to model compositionality by regression: for example, Baroni-Zamparelli and Guevara extracted context vectors from corpora not only for the components *v1* and *v2*, but also for the composed expression *v3* (i. e., they extracted contextual representations for *beautiful* and *dancer*, but also for the observed pair *beautiful_dancer*). Starting from these data, Guevara built a model of Adjective-Noun compositionality through a

---

[144]See, for example, T. A. Plate, *Holographic reduced representations: convolution algebra for compositional distributed representations,* in *Proceedings of the 12th International Joint Conference on Artificial Intelligence,* Sydney (AUS), 1991, pp. 30-35. More recent works using a circular convolution-based approach are: M. N. Jones, D. J. K. Mewhort, *Representing word meaning and order information in a composite holographic lexicon*, in *Psychological Review,* vol. 114, 2007, pp. 1-37; D. Widdows, *Semantic Vector Products: some initial investigations*, in *Second AAAI Symposium on Quantum Interaction,* Oxford (UK), 2008; E. Giesbrecht, *In search of semantic compositionality in Vector Spaces*, in *Proceedings of International Conference on Computational Science*, Moscow (RUS), 2009, pp. 173-184; E. Grefenstette *et al.*, *Concrete sentence spaces for compositional distributional models of meaning,* in *Proceedings of the 9th International Conference on Computational Semantics,* 2011, pp. 125-134.

[145] See J. Mitchell, M. Lapata, *ibid.*, 2008; J. Mitchell, M. Lapata, *ibid.,* 2010.

[146]See E. Guevara, *ibid.*, 2011.

supervised machine-learning approach based on partial least squares regression (PLS): the aim of his method is to learn the transformation function that best approximates *v3* on the basis of *v1* and *v2*. Similarly, Baroni and Zamparelli assumed that each adjective "corresponds" to a linear transformation function: they modeled the adjective-noun compositionality by approximating *v3* only on the basis of the noun, and running a different regression analysis for each adjective in the data.

This kind of supervised learning can be seen as a way of optimize the weight parameters of the compositional function through linear regression. 147

Another possible approach to the problem is based on kernel methods.

One of the most common tests for the models of word meaning in context is the formulation of appropriate paraphrases, because paraphrases typically apply to some senses of a word, not to all.

Vector space models can predict the rightness of a paraphrase measuring the similarity between vectors, but this task can be addressed with kernel methods, which project sets of items into implicit feature spaces for similarity computation. Even if both models are used for tasks regarding similarity, they focus on different types of information: most of the current kernel methods compare syntactic structures, and introduce semantic information

[147]See M. Baroni, R. Zamparelli, *Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space*, in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, Association for Computational Linguistics, East Stroudsburg (Pennsylvania), 2010, pp. 1183-1193; E. Guevara, *A regression model of adjective-noun compositionality in Distributional Semantics,* in *Proceedings of the 2010 Workshop on Geometrical Models of Natural Language Semantics*, Association for Computational Linguistics, Uppsala (SWE), 2010, pp. 33-37; E. Guevara, *ibid.*, 2011.

only in a second step, in order to smooth syntactic similarity; instead, vector space models try to model the interaction between the lexical meanings of the words composing a complex linguistic expression. 148

4.2 Vector space models for word meaning in context

Another family of approaches deals with the problem of compositionality in a slightly different manner: instead of focusing on the process of composition, trying to identify the function (or the functions) corresponding to the best approximation of the way in which the meanings of the components interact in the resulting complex expression, they concentrate on the computation of the meaning of individual words in context.

We have seen in the preceding chapters that the meaning of the words can vary substantially between occurrences, because of the influence of the context 149; we know that words can be polysemous, and in such cases we usually have to select a sense from an inventory of possible ones. But which senses should we include in this dictionary?

Adam Kilgarriff's criticism of the notion of "word sense" showed its lack of theoretical foundation. Since the problem of Word Sense Disambiguation could be seen as a classification task (we have to

[148]See, for example, A. Moschitti, S. Quarteroni, *Kernels on linguistic structures for answer extraction*, in *Proceedings of the Association of Computational Linguistics: Human Language Technologies,* Association for Computational Linguistics, Columbus (Ohio), 2008, pp. 113-116.

[149]See paragraphs 1.4 and 2.3.

assign a label, corresponding to one of the possible senses, to every word occurrence), where it is even difficult even to enumerate all the classes and to trace clear distinctions between them (without a definition for the notion of "word sense", we cannot know how many senses a word has, and even if we could say a number based on our intuition, the boundaries between word senses would not be clear-cut), it is not surprising that this task has always been one of the most challenging in NLP.

Some NLP studies in the second half of the 90s have showed that, in order to disambiguate word senses, there is no need to define their inventory *a priori*: using disambiguation algorithms based on agglomerative hierarchical clustering 150, senses can
be constructed as abstractions over clusters of similar contexts of the ambiguous word.
For example, in (Schütze: 1998) first-order vector representations of word meaning are collected by using co-occurrence counts from the entire corpus; then, second-order representations are computed for individual word istances in their respective contexts, by summing up all first-order vectors of the words in the context. The resulting

[150]Hierarchical clustering is a data mining technique which aims at building a hierarchy of clusters.

In agglomerative hierarchical clustering, in particular, we follow a bottom-up approach: in the starting situation, every observation is in its own cluster, and pairs of clusters are gradually merged on the basis of same criterion of similarity. At the end of the process, only one cluster remains, including all the observations in the dataset.

In our case, the observations are words, represented through their co-occurrence vectors, and their similarity is measured as the distance between the vectors.

See P. N. Tan *et al., Introduction to Data Mining*, Pearson Addison Weasley, Boston (Massachusetts), 2006, in particular chapter 8; see also http://en.wikipedia.org/wiki/Hierarchical_clustering.

clusters correspond to different word senses. 151

These models generally aim at the integration of a wide range of contextual information, ignoring syntactic structures. Other models take into account more restricted contexts and try to model specific phenomena of contextual modification: for example, the cited work of Kintsch focuses on the predicate-argument combination (consequently, the context consists of a single word), and uses vector representations of the predicate *p* and of the argument *a* to identify a set of words that are similar to both; the meaning of the composed expression is computed as the centroid of the vectors of this set. Even in this kind of model, the syntactic relations are not considered. 152

In their works, Erk and Padó criticize this shortcoming, which has at least two negative implications for the VSMs of compositionality:

- they fail to capture differences of meaning between sentences that are due to differences in relation. For example, see the sentences:

  4.1 A girl is drawing.
  4.2 He's drawing a girl.

In the first sentence, the phrase *a girl* is the subject, while in the

[151]See, for example, D. Yarowsky, *Unsupervised word sense disambiguation rivaling supervised methods,* in *Proceedings of the 33rd Annual Meeting of the Association of Computational Linguistics,* Association for Computational Linguistics, Cambridge (Massachusetts), 1995; J. O. Pedersen, H. Schütze, *ibid.,* 1995; H. Schütze, *Automatic word sense discrimination,* in *Journal of Computational Linguistics,* MIT Press, Cambridge (Massachusetts), vol. 24, no. 1, 1998, pp. 97-124.
[152]W. Kintsch, *ibid.*, 2001.

second it is the object.

A VSM ignoring the syntactic relation between *a girl* and *draw* will not be able to identify the difference of meaning between the two sentences;

- there is no upper limit to the length of a sentence, and therefore to the amount of structural information to be encoded. Most of the vector composition methods result in a single vector, representing the meaning of a complex linguistic expression, but a single vector is probably not enough to encode such a complexity.

The vector space model introduced in (Erk and Padó: 2008) is called *structured* because the argument structure is explicitly represented, using multiple vectors for each word.

The basic intuition of their model is that the contextual interpretation of a word is often guided by expectations about typical events: in a sentence like *listen to music*, the verb will be interpreted to match typical action that can be performed with music, while the substantive will be interpreted in order to match the expectations about things that can be listened. [153]

In linguistics, expectations have been used in semantic theories in the form of *selectional restrictions* and *selectional preferences* 154; more recent studies have also aimed at the extraction of selectional

[153]K. Erk, S. Padó, *ibid.*, 2008.

[154]J. J. Katz, J. Fodor, *The structure of semantic theory*, in *The structure of language*, Prentice-Hall, 1964; Y. Wilks, *Preference semantics*, in *Formal semantics of Natural Language*, Cambridge University Press, Cambridge (UK), 1975.

Selectional restrictions / preferences are semantic constraints on arguments of a verb: for example, the verb *catch* can have as direct object only things that can be caught. See C. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing,* MIT Press, Cambridge (Massachusetts), 1999, chapter 8.

preferences from corpora 155, and some of them have proposed vector space models computing the typicality of an argument through similarity to previously seen arguments. 156

Instead of representing word meaning with a single vector, Erk and Padó encoded each word as a combination of the following elements:

- a vector corresponding to the lexical meaning of the word;
- a set of vectors, each representing the selectional preferences for a particular syntactic relation supported by that word.

In Figure 1 we can see an example: the verb *catch* is represented by a lexical vector (the central square), while the three arrows link it to its preferences for its subjects (*he, fielder, dog*), for its objects (*cold, baseball, drift*) and for the verbs for which it appears as a complement (*accuse, say, claim*). Erk and Padó's representation includes both selectional preferences and inverse selectional preferences ($subj^{-1}$, obj^{-1}, $comp^{-1}$). 157

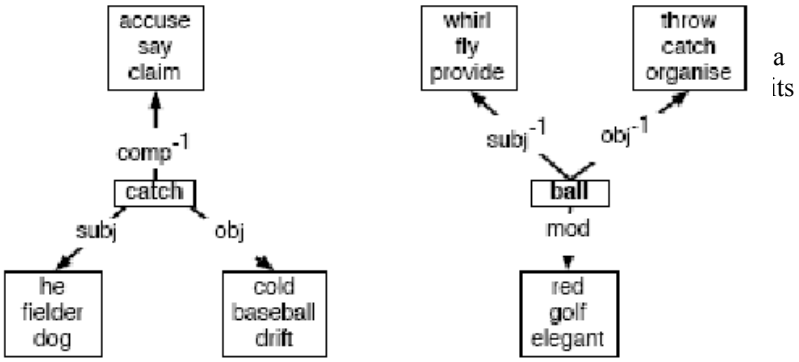
---

[155]See P. Resnik, *Selectional constraints: an information-theoretic model and its computational realization*, in *Cognition*, vol. 61, 1995, pp. 127-159; C. Brockmann, M. Lapata, *Evaluating and combining approaches to selectional preference acquisition*, in *Proceedings of the European Chapter of the Association for Computational Linguistics,* Association for Computational Linguistics, 2003, pp. 27-34.

[156]K. Erk, *A simple, similarity-based model for selectional preferences*, in *Proceedings of the Association for Computational Linguistics,* Association for Computational Linguistics, 2007, pp. 216-223; S. Padó, M. Lapata, *Dependency-based construction of semantic space models*, in *Computational Linguistics,* vol. 33, no. 2, 2007, pp. 161-199.

[157]*Inverse selectional preferences* can be defined as selectional preferences of arguments for their predicates, such as the preference of a subject or object for its verb.

See K. Erk *et al.*, *A flexible, corpus-driven model of regular and inverse selectional preferences,* in *Journal of Computational Linguistics,* MIT Press, Cambridge (Massachusetts), vol. 36, no. 4, 2010, pp. 723-764; H. Jang, J. Mostow, *Inferring selectional preferences from part-of-speech N-grams,* in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics,* Association for Computational Linguistics, Stroudsburg (Pennsylvania), 2012, pp. 377-386.

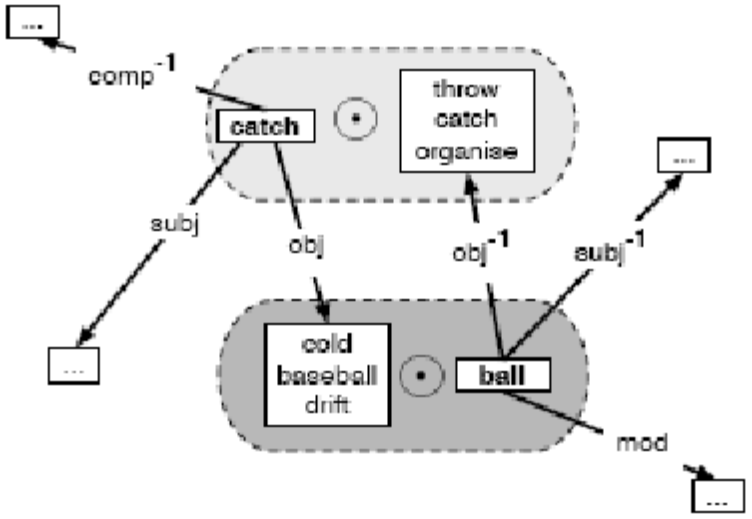


In this structured vector space, the computation of word meaning in context, given a predicate *p* and an argument *a*, can be carried out through:

- the combination of the lexical vector of *p* with the inverse preference vector of *a* for their syntactic relation *r* (for example, in Figure 2: the lexical vector of *catch* is combined with the inverse object preference vector of *ball*);
- the combination of the lexical vector of *a* with the preference vector of *p* for their syntactic relation *r* (in Figure 2, the lexical vector of *ball* is combined with the object preference vector of *catch*).

As pointed out by the authors, their model can be considered within the framework proposed by Mitchell and Lapata: they have encoded the selectional preferences of the two original vectors as additional knowledge K. 158

[158]See formula 4 in paragraph 4.1.



In another study, Erk and Padó proposed an exemplar-based model for word meaning in context 159. In cognitive science, while prototype models predict the degree of membership of a concept to a category on the basis of its similarity to a single prototype for that category, exemplar theory represent categories as collections of previously met exemplars: our category judgements are based on the comparison of every new instance with many stored memories of that category. 160

Erk and Padó's model represents each target word as a set of exemplars, where an exemplar is a vector representation of a sentence in which the target occurs. They dealt with the problem of polysemy by activating only relevant exemplars in a given sentence

[159]K. Erk, S. Padó, *Exemplar-based models for word meaning in context*, in *Proceedings of 2010 Conference of the Association for Computational Linguistics,* Association for Computational Linguistics, Uppsala (SWE), 2010, pp. 92-97.
[160]G. L. Murphy, *The big book of concepts,* MIT Press, Cambridge (Massachusetts), 2002.

context, i.e. the exemplars whose similarity score with respect to the sentence context *s* was over a determined threshold.

The authors applied this approach in a paraphrasing task: given a target word *w* in a context *s*, and given a list of potential paraphrases of *w,* the task is to predict which paraphrases are appropriate for *w* in *s*. Since paraphrases are typically applicable to only a particular sense of a word, they computed the similarity of *w* in *s* with the vector representations of its exemplars, in order to activate only those whose similarity with the target in context exceeded the threshold (the relevant exemplars). On the same basis, the model ranked the final list of paraphrases by their goodness of fit to *w* in *s*.

We presented these models separately, because they are probably more suitable for the purposes of Sentiment Analysis: we do not aim to build general models of compositionality, we just need to handle the syntactic phenomena causing polarity shifts 161; similarly these models, instead of aiming to an exhaustive coverage of all kinds of meaning composition, focus on a more specific problem, i. e. the disambiguation of the word meaning in context.

[161]See paragraph 1.4.

The topic of contextual polarity has been widely discussed in L. Polanyi, A. Zaenen, *ibid.*, 2004; M. Taboada *et alii*, *ibid.*, 2011.

4.3 Sentiment Analysis and Compositionality

Now we have introduced the most recent work on Distributional Semantics and Compositionality; some of the studies presenting computational approaches for Sentiment Analysis in context have been presented in the past chapters.

It is worth to recall briefly at least two of them:

- the methods presented by Choi and Cardie, based on a sentiment lexicon including both positive words, negative words and negators (they were the first to consider even content words as potential negators) and on the introduction of composition rules, applied once a particular syntactic pattern is detected, through which it is possible to calculate the contextual polarity of phrases and sentences; 162
- the approach introduced by Socher *et al.*, based on recursive autoencoders for sentence-level prediction of sentiment label distribution; interestingly, in a further study they also used a similar recursive neural network model for semantic compositionality. [163]

As far as I am concerned, no one has ever applied a distributional approach to the problem of contextual polarity, that is to say no one

[162]See C. Cardie, Y. Choi, *Learning with compositional semantics as structural inference for subsentential Sentiment Analysis*, in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008*), Waikiki (Hawaii), 2008. A further development of this work is A. Yessenalina, C. Cardie, *Compositional matrix-space models for Sentiment Analysis,* in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011*), Edinburgh (SCO), 2011, pp. 172-182.

[163]See R. Socher *et al.*, *ibid.*, 2011; R. Socher *et al.*, *Semantic Compositionality through recursive matrix-vector spaces,* in *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP 2012*), Jeju (KOR), 2012.

has ever thought of the composition of the meaning of word with the meanings of the other words in its context as an operation which relocates the corresponding vector in the distributional space. Since in the preceding chapter we built prototypical vectors for words with a positive / negative polarity, it could be interesting to verify if the movements of the vectors in our distributional space, due to semantic composition, correspond to the shifts that we perceive in the semantic orientation of a composed expression.
In the following paragraphs, we will conduct some experiments on sentiment compositionality inspired to Mitchell-Lapata's works

4.4 An experiment of "sentiment composition"

In our experiment, we focused on a single syntactic pattern, i. e. verb + object.
We selected 50 verbs, 25 for each polarity, which were correctly classified in the preceding task, using the prototypes built with Turney and Littmann's seed set.
We associated to every verb a triple of its possible objects: the first two keep the typical polarity of that verb, while the last one may invert it (or, at least, it makes the composed expression more "neutral").
Using different compositional models, we generated three V-N pairs for each verb, combining it with the three arguments of its triple. Then, we measured the distance of each pair from the prototypes: we expect the pairs generated by the combination of the verb with one of the first two arguments to maintain its prior polarity, while

the pair generated by the combination with the last argument may produce a polarity shift and make the composed vector move towards the opposite prototype.

We rated the sentiment of the V-N pairs with native speakers judgements collected with Crowdflower. 164

Of course, not all the triples were equally correct. But, in general, we can observe a polarity shift of the third compound, compared with the other two. Moreover, since the number of the triples is limited, we could observe whether there is a significant correlation between the sharpness of the judgment of the subjects and the ability of our model to classify it.

First, we extracted from Distributional Memory a matrix containing all the words involved in our composition experiment (the words of the dataset, plus Turney and Littmann's seed sets): only three triples, since two out of their three arguments were not present in Distributional Memory, were discarded.

Then, we built the composition vectors using two different models:

- the **additive model**, building the vector of a composed expression as the sum of the words composing it (see eq. 1, paragraph 1);
- the **multiplicative model**, building the vector of a composed expression as the vector-pointwise multiplication of the words composing it (see eq. 2, paragraph 1).

After building the composed vectors, combining every verb with its three arguments,

we measured their distance from Turney-Littmann's prototype

[164]The table with the ratings of the triples can be consulted in the appendix.

vectors, both in the standard DM vector space (30686 dimensions) and in vector spaces whose dimensionality had been reduced through Random Indexing. 165

For each model we evaluated:

- the accuracy in the polarity classification of composed expressions;
- the accuracy in the recognition of a polarity shift. A polarity shift was correctly recognized when:

a) *the compositions resulting from the combination of the verb with the first two arguments maintain the prior polarity of the verb and*

b) t*he composition resulting from the combination of the verb with the third argument invert the prior polarity of the verb.*

The total number of the triples was 47, while the number of the compositions was 141.

The results of the experiment can be observed in the following tables.

[165]See M. Sahlgren, *ibid.,* 2005.

ADDITIVE MODEL

Space	Compositions	Complete triples	Compositions correctly classified	Polarity shifts correctly identified
Standard DM (30686 dimensions)	141	47	83 (58,86%)	13 (27,65%)
Reduced to 10000 dimensions	141	47	73 (51,77%)	5 (10,6%)
Reduced to 5000 dimensions	141	47	74 (52,48%)	10 (21,27%)

MULTIPLICATIVE MODEL

Space	Compositions	Complete triples	Compositions correctly classified	Polarity shifts correctly identified
Standard DM (30686 dimensions)	38	10 [166]	26 (68,4%)	6 (60%)
Reduced to 10000 dimensions	134	44	73 (54,47%)	16 (36,36%)
Reduced to 5000 dimensions	132	44	77 (58,3%)	13 (29,5%)

[166]Since the data of this matrix were extremely sparse, we considered as complete a triple even though it has only two compositions that have not been zeroed, provided that one of the two remaining compositions is the one shifting the polarity.

A first consideration: all the models seem to have a pretty good performance as regards the polarity classification of the compositions (the accuracy never drops below 51%), but this is principally due to the way we constructed our dataset; indeed, for our compositions we selected verbs that had been correctly classified in our preceding task and we combined them, at least in two thirds of the cases, with arguments preserving their prior polarity. We found out that preserving the prior polarity of the verb was the normal tendency, even when the argument should have changed it, and this was probably one of the main typologies of misclassification error. Assumed this, a globally positive accuracy for the polarity classification of the compositions should not surprise us.

More interesting was the accuracy in the recogition of the polarity shifts, that is to say the ability of the model to predict a modification of the polarity caused by a particular argument. From this point of view, the results are much worse:

- all the additive models had very low performances and, in general, seem not to be able to recognize the polarity shifts;
- multiplicative models's performances were considerably better, and this is consistent with the findings of Mitchell and Lapata. The main problem of this models is that they produce extremely sparse vectors, because many components are multiplied by 0; many vectors have been totally zeroed, so that they were no more comparable.

One of our models, the simple multiplicative model, was afflicted by the resetting of almost all the vectors in the dataset. The results, for the few remaining data, seemed promising: a 68% accuracy in

the classification of the compositions, and a 60% accuracy in the prediction of the polarity shifts. For example, the model was able to identify the polarity shifts of generally positive verbs like *desire* and *excite* when they take a negative argument. But the shortage of the remaining data does not allow to formulate any generalization from the results of this model.

The other multiplicative models used spaces whose number of the dimensions had been reduced through Random Indexing, so that they did not suffer for the same sparsity of data. Their performances were equally much better than those of the additive models: among the models tested on the (almost) complete dataset, the multiplicative one with 10000 dimensions was the best in recognizing the polarity shifts (36,36%).

As we anticipated, our results are consistent with the findings of Mitchell and Lapata: additive models are always outperformed by multiplicative models. Of course, they are "naive" models because:

- they are symmetric, and thus they do not take word order into account and make no distinction between the combined constituents;
- they are essentially bag-of-words models of compositions, in the sense that they assign the same representation to any sentence containing the same constituents;
- even if some information about syntactic relation can be introduced, for example assigning different weights to the contribution of each constituent, their representations cannot be said to have internal structure.

In spite of this, and even in their simplest form, their performances are pretty good.

Additive models, in addition to the same limits of multiplicative ones, blend together the content of all words involved to produce something in between them all 167, but we would rather like a model of semantic composition that selects and modifies particular aspects of the words involved.

We think that multiplicative models reach this aim, because their representations preserve only the dimensions that are common to all the words composing a complex expression, while the other dimensions are filtered out. The excluded contexts correspond to dimensions of meaning that are not relevant for a certain semantic composition: if they were relevant, they would be shared by all its components. Consequently, we observe a noise reduction in the resulting composition.

Finally, it would be interesting to try a totally different approach to address the problem of detecting polarity shifts, for example a semantic composition model similar to Erk and Padò's: given a predicate-argument relation, we could compose the vector of the predicates with the inverse selectional preferences of the arguments, and see whether the repositioning of the vectors reflects the expected preservation / inversion of the polarity of the composed expression. But we will leave this for a future research.

[167]See J. Mitchell, M. Lapata, *ibid.*, 2011.

Conclusions

In this study, we have presented some simple distributional approaches to sentiment-related tasks.

After showing the state of the art in the areas of research of our interest, Sentiment Analysis and Distributional Semantics, we searched for possible, promising intersections between these two fields. The idea on which our work is based is that, in order to recognise the "polarity" (positive or negative) of a linguistic expression, it is possible to extend Firth's claim as follows: "you shall know the polarity of a word from the company it keeps".

In the first experiment, we tested this possibility in a simple task of polarity classification, in which the aim was to identify the polarity of the words of a sentiment lexicon.

Starting from Turney and Littmann's work, we used the resources of the Distributional Memory framework to build a semantic vector space and to classify the words on the basis of their distance from prototype vectors [168]. We built different prototype vectors, using various seed sets, but the best performance has been achieved with Turney-Littmann's seed set, which was composed of common adjectives.

Moreover, we tried an experiment of "sentiment compositionality", using additive and multiplicative models similar to those recently proposed by Mitchell and Lapata, and our results are consistent with

[168] P. D. Turney, M. Littmann, *ibid.*, 2003.

their findings [169]: multiplicative models seem to be more suitable for semantic composition, even for a polarity classification task, since they preserve only the dimensions of meaning that are shared by all the words composing a more complex linguistic expression. In a future study, it would be interesting to test a model of "sentiment composition" implementing knowledge about selectional preferences, following Erk and Padó's approach. [170]

In our opinion, there is another research area in which distributional methods could be productively applied, that is the identification and extraction of the features of general concepts and items. The identification of an object's features allows a more fine-grained Sentiment Analysis, because it is possible to identify those aspects about which people are expressing judgements. But it is worth stressing that there is a growing interest in this task, even outside the borders of this discipline, for example in cognitive sciences. Some theories on the representation of concepts in the human brain have supposed a feature-based model, in the sense that such representations can be thought as patterns of activation over sets of interconnected semantic feature nodes [171]. The aim of a lot of recent studies is to develop computational semantic models for the extraction -from large textual corpora- of salient properties characterizing entities, in order to test their ability to learn the inner

[169]J. Mitchell, M. Lapata, *ibid.*, 2008; J. Mitchell, M. Lapata, *ibid.*, 2011.
[170]K. Erk, S. Padó, *ibid.,* 2008.
[171]G. Cree, K. McRae, *Analyzing the factors underlying the structure and computation of the meaning of* chipmunk, cherry, chisel, cheese *and* cello *(and many other such concrete nouns)*, in *Journal of Experimental Psychology*, vol. 132, no. 2, 2003, pp. 163-201; K. McRae *et al.*, *Semantic feature production norms for a large set of living and nonliving things,* in *Behavioral Research Methods, Instruments and Computers,* vol. 37, 2005, pp. 547-559.

structure of concepts. [172]

It would be very promising, for the purposes of Sentiment Analysis, to combine pattern-based and distributional methods to extract the main features of objects and entities that are cited in online texts, in particular those features which tend to co-occur with subjectivity clues: for instance, if a feature occurs very frequently in the reviews of a certain category of products and it is often associated with verbs / adjectives expressing opinions or subjective evaluations, it is highly probable that the users consider that feature important in judging that kind of product. Of course, a system able to identify such features would be of incredible interest for web marketing, just to make an example.

Investigating sentiment with distributional methods is an intriguing perspective, because it creates a link -and consequently new possibilities of an exchange of methods and resources- between theoretical researches on language and cognition and a subfield of the studies in knowledge discovery that is expected, with the development of Web 2.0 and the increasing availability of opinionated textual data, to take on an important role in the marketing and in the public communication of the future.

Of course, there is still much to do: first of all, we are just at the beginning regarding the attempts of computational modeling of the

[172]See, for example, M. Baroni, A. Lenci, *Concepts and properties in word spaces*, in *Rivista di Linguistica*, vol. 20, n. 1, 2008, pp. 53-86; M. Baroni *et al.*, *Strudel: a corpus-based semantic model based on properties and types*, in *Cognitive Science*, vol. 34, n. 10, 2010, pp. 222-254; C. Kelly *et al.*, *Acquiring human-like feature-based conceptual representations from corpora*, in *Proceedings of the NAACL HLT 2010 - 1st Workshop on Computation Neurolinguistics*, Los Angeles, 2010, pp. 61-69; C. Kelly *et al.*, *Semi-supervised learning for automatic conceptual property extraction*, in *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2012)*, Montreal, 2012, pp. 11-20.

processes of meaning composition; consequently, we are very far from the realization of a syntax-aware model for the calculation of the contextual polarity of phrases and sentences. In this thesis work, we just tried to lay the foundations for a future in-depth research.

APPENDIX

Distributional Memory Perl Scripts

1) filter_by_field.pl

```
# This script, freely downloadable from the web page
of Distributional Memory, can be
# used to filter the words of a matrix, extracting
only the elements of interest: the user just # needs
to specify, in a separate file, a "white list" of
target elements and an attribute, and # the only
tuples extracted will be those whose value for that
attribute is present in the
# list. [173]


#!/usr/bin/perl

use Getopt::Std;

{
$usage = <<"_USAGE_";

Usage:

filter_by_field.pl [-s] [-fN] target_list
to_be_filtered > filtered
filtered_by_field.pl -h

- target_list is in one-string per line format;
```

[173] http://clic.cimec.unitn.it/dm/

```
- to_be_filtered is a list where each line has the
same number of
  fields (space or tab delimited).

The script keeps only those lines of to_be_filtered
where the element
in field N (1 by default, or specified by -f option)
is identical to
(or, if -s option is passed, is NOT identical to)
one of the strings
in target_list.

This script is free software. You may copy or
redistribute it under
the same terms as Perl itself.

_USAGE_
}

{
    my $blah = 1;
# this useless block is here because here document
confuses emacs
}

%opts = ();
getopts('sf:h',\%opts);

# The getopts function takes, as the first
parameter, a string containing all the possible  #
arguments of the script (s and h are boolean flags,
f: takes an argument).
# %opts is a hash table containing the values of the
```

```
arguments: if h is passed, the
# instructions for the use of the script are
printed.

if ($opts{h}) {
    print $usage;
    exit;
}



$stop_op = 0;
if ($opts{s}) {
  $stop_op = 1;
}

$target_field = 0;
if ($opts{f}) {
    $target_field = $opts{f} - 1;
}

if (!(open WLIST, shift)) {
    print $usage;
    exit;
}


# If either the first or the second file cannot be
opened, the variable $usage is printed.


while (<WLIST>) {
    chomp;
    s/\r//g;
    $in{$_} = 1;
}
```

```
close WLIST;

# The command chomp and regular expression  s/\r//g
remove all the newlines and the  # return
characters.
# The function reads from <WLIST> one line at a
time. For every target element in the  # first file,
a hash function sets a flag to 1, and associates it
to a key (the target element).


if (!(open ULIST, shift)) {
    print $usage;
    exit;
}

while (<ULIST>) {
    $input = $_;
    chomp;
    s/\r//g;
    @F = split "[\t ]",$_;
    if ($stop_op) {
    if ($in{$F[$target_field]}) {
        next;
    }
    }
    else {
    if (!$in{$F[$target_field]}) {
        next;
    }
    }
    print $input;
}
```

```
close ULIST;


# In the case the value of $stop_op is 1 (the user
has passed the option -s, and                 #
consequently only the tuples whose values for the
attribute $target_field are not
# present in the WLIST should be preserved), the
function verifies the value of the hash
# $in for the input string in the specified field as
a key: if the value is 1 (i.e. the value of # the
attribute $target_field, for that tuple, equals the
input string), the function will
# pass to the next iteration and the tuple will be
discarded.
# In the case the value of $stop_op is 0 (only the
tuples whose values are present in the
# WLIST should be preserved), the function verifies
the value of the hash
# $in for the input string in the specified field as
a key: if the value is 0 (i.e. the value of # the
attribute $target_field, for that tuple, does not
equal the input string), the function
# will pass to the next iteration and the tuple will
be discarded.
```

2) build_matrix_from_tuples.pl

```
# After the extraction of the elements of interest
from the Distributional Memory tensor,
# this script can be used for building matrices from
tuples of the format:
# element 1 element 2 score.

#!/usr/bin/perl -w

use strict "vars";
use Getopt::Std;

my $usage;
{
$usage = <<"_USAGE_";
This script takes as input a prefix string and a tuple
file in (tab-
or space-delimited) format:

row-element column-element score

and creates the 2 files prefix.mat and prefix.col,
where prefix is the
input prefix string (existing files with these names
are deleted).

The prefix.mat file contains a matrix where each row
corresponds to a
row element in the input tuples, the first field
contains the row
element and all the other tab-delimited fields of the
row are filled
by the scores of that row element with each of the
```

column elements (if
a row element did not occur with a column element, the corresponding
field gets a 0 score). The prefix.col file contains the column
elements, one per line, in the order in which they are represented in
the matrix (both row and column elements are in ASCII-based dictionary
order).

Suppose for example that the file input.txt contains:

```
dog tail 23
cat tail 21
dog barks 15
```

and that we call the script as:

```
build_matrix_from_tuples.pl output input.txt
```

This will generate:

1) an output.mat file containing the following lines:

```
cat 0 21
dog 15 23
```

2) an output.col file containing the following lines:

```
barks
tail
```

```
Usage:

build_matrix_from_tuples.pl -h

build_matrix_from_tuples.pl prefix input


Copyright 2009, Marco Baroni

This program is free software. You may copy or
redistribute it under
the same terms as Perl itself.

_USAGE_
}
{
    my $blah = 1;
# this useless block is here because here document
confuses emacs
}

my %opts = ();

getopts('h',\%opts);

if ($opts{h}) {
    print $usage;
    exit;
}


 # The getopts function takes, as the first
```

```
 parameter, a string containing all the possible  #
 arguments of the script (h is a boolean flag).
  # %opts is a hash table containing the values of the
arguments. If the parameter h is
  # passed, the instructions for the use of the
function will be printed.


my $prefix = shift;
my $table = shift;

my $columns = $prefix . ".col";
my $matrix = $prefix . ".mat";

if (-e $columns) {
    print STDERR "$columns already exists, deleting
previous version\n";
    `rm -f $columns`;
}
if (-e $matrix) {
    print STDERR "$matrix already exists, deleting
previous version\n";
    `rm -f $matrix`;
}


# The function checks whether a .mat and/or a .col
files with the same name are already
# existing or not. In the first case, the files will
be replaced by their new version.


my %seen = ();
my @col_els = ();
```

```
my @row_els = ();
my %score_of = ();
my $row_el;
my $col_el;
my $score;
open TABLE,$table;
while (<TABLE>) {
    chomp;
    ($row_el,$col_el,$score) = split "[\t ]+",$_;

# The triples are splitted and the values are saved in
three different arrays

    if (!($seen{"col"}{$col_el}++)) {
     push @col_els, $col_el;
    }
    if (!$seen{"row"}{$row_el}++) {
     push @row_els, $row_el;
    }

# If the elements in the row (first field) / in the
column (second field) -respectively the
# current values of $col_el and $row_el- have not been
seen yet, they will be pushed
# respectively in the hash tables @col_els and
@row_els.

    $score_of{$row_el}{$col_el} = $score;
}
close TABLE;

# The score is saved in another hash table, using the
row-column values as keys.

%seen = ();
my @sorted_col_els = sort @col_els;
```

```
@col_els = ();
my @sorted_row_els = sort @row_els;
@row_els = ();

# The row and the columns are then sorted.

open COLUMNS,">$columns";
foreach $col_el (@sorted_col_els) {
    print COLUMNS $col_el,"\n";
}
close COLUMNS;

# Every value in @sorted_col_els is printed in
$columns

open MATRIX,">$matrix";
foreach $row_el (@sorted_row_els) {
    print MATRIX $row_el;
    foreach $col_el (@sorted_col_els) {
     if (!($score = $score_of{$row_el}{$col_el})) {
         $score = 0;
     }
     print MATRIX "\t",$score;
    }
    print MATRIX "\n";
}
close MATRIX;


# The function prints, for every line of the output
file:
# - every value in @sorted_row_els;
# - if a value exists, in the hash table $score_of for
```

```
the keys $row_el and $col_el, that
# value will be printed; otherwise, 0 will be printed.
```

3) sum_vectors.pl

```
# This script is particularly useful for building
prototypes of set of vectors.

#!/usr/bin/perl -w

use strict "vars";
use PDL;
use Getopt::Std;

my $usage;
{
$usage = <<"_USAGE_";
This script takes two input files, one with element-
set pairs, the
other with vectors representing the elements, and
returns, for each
set in the first file, a sum of the vectors of the
elements in the
set.

The element-set file has one (tab- or space-delimited)
element-set
pair per line (an element can occur with more than one
vector). Lines
in the vector file are also tab- or space-delimited,
have a first
field with an element, and the values of the vector
```

representing the
element in the remaining field (it is assumed that all vectors have
the same number of dimensions). Vectors for elements that are not in
any set are ignored, and if an element in a set does not have a vector
in the vector file, it is also ignored (if a set is only made of
elements not in the vector file, it will end up having a zero vector
in the output).

If the option *-n* is passed, the vectors are normalized before summing.

The (tab-delimited) output has one set per line, followed by its
summed vector.

NB: The script requires PDL to be installed.

Usage:

```
sum_vectors.pl -h

sum_vectors.pl target-set-file vector-file > outfile

sum_vectors.pl -n target-set-file vector-file >
outfile
```

```
This program is free software. You may copy or
redistribute it under
the same terms as Perl itself.

_USAGE_
}
{
    my $blah = 1;
# this useless block is here because here document
confuses emacs
}

my %opts = ();
getopts('hn',\%opts);

if ($opts{h}) {
    print $usage;
    exit;
}

my $normalize = 0;
if ($opts{n}) {
    $normalize = 1;
}


 # The getopts function takes, as the first
 parameter, a string containing all the possible  #
 arguments of the script (s and h are boolean flags,
 f: takes an argument).
 # %opts is a hash table containing the values of the
 arguments: if h is passed, the
 # instructions for the use of the script will be
printed.
```

```
my $element_set_file = shift;
my $vector_file = shift;


my %set_list_of_element = ();
my %seen_pairs = ();
my %is_set = ();
open FILE1,$element_set_file or
    die "could not open $element_set_file";
while (<FILE1>) {
    chomp;
    s/\r//g;
    my ($element,$set) = split "[\t ]+",$_;
    if ($seen_pairs{$element}{$set}) {
     next;
    }
    push @{$set_list_of_element{$element}},$set;
    $seen_pairs{$element}{$set} = 1;
    $is_set{$set} = 1;
}


 # The command chomp and regular expression  s/\r//g
 remove all the newlines and the
 # return characters.
 # The function reads from <WLIST> one line at a
 time. The lines are splitted and the
 # element and the set are saved in the arrays
 $element and $set.
 # The elements of $set are saved in the hash table
 $set_list_of_element, accessible
 # through the key $element. When a new set $set is
 seen, the value of the hash table
 # $is_set for the key $set is set to 1; when a new
 pair $element - $set is seen, the value
 # of the hash table $seen_pairs for the keys
 $element and $set is set to 1.
 # If the pair element-set has already been seen, the
 function will pass to the next
 # iteration.
```

```
close FILE1;
%seen_pairs = ();

my %summed_vector = ();
my $d = 0;
open FILE2, $vector_file;
while (<FILE2>) {
    chomp;
    s/\r//g;
    my @F = split "[\t ]+",$_;
    if (!$d) {
     $d = $#F;
     # this is the number of dimensions, since there
is
     # one extra item (the element)
    }
    my $element = shift @F;

    if (!defined($set_list_of_element{$element}[0])) {
     next;
    }
    foreach my $set
(@{$set_list_of_element{$element}}) {
     if ($normalize) {
         my $temp_raw =  pdl @F;
         my $temp_length = sqrt(sum($temp_raw**2));
         if ($temp_length == 0) {
           $summed_vector{$set} += $temp_raw;
         }
         else {
           $summed_vector{$set} += $temp_raw /
$temp_length;
```

```
        }
        undef $temp_raw;
    }
    else {
        $summed_vector{$set} += pdl @F;
    }
    }
}
close FILE2;


# If the option -n for the normalization is passed,
the vector to be summed is saved in
# $temp_raw, while the norm of the vector is saved in
$temp_length: if the norm of the
# vector is 0, $temp_raw will be directly summed to
$summed_vectors {$set}, otherwise
# $temp_raw will be first divided by its norm..
# If the option -n is not passed, the vector will be
directly summed to $summed_vectors
# {$set}.
# The operation is repeated for each of the specified
vector sets.

foreach my $set (keys %is_set) {
    if (!defined($summed_vector{$set})) {
    $summed_vector{$set} = zeroes($d);
    }
    print $set;
    my @temp_array = swcols($summed_vector{$set});
    foreach my $value (@temp_array) {
    if ($value == 0) {
        print "\t",$value;
```

```
    }
    else {
        printf "\t%.4f",$value;
    }
    }
    print "\n";
    undef @temp_array;
}

# Foreach of the sets, if $summed_vector{$set} is not
defined, all the values of
# $summed_vector{$set} are set to 0.
# In @temp_array, by using the function swcols, an
array of strings is saved.
# Then, each value of @temp_array is checked, and
different formats for printing are
# specified.
```

4) compute_cosines_of_pairs.pl

```
# This script can be used for computing vector
similarity.

#!/usr/bin/perl -w

use strict "vars";
use PDL;
use Getopt::Std;

my $usage;
{
```

```
$usage = <<"_USAGE_";
This script takes as input a list of (tab- or space-
)delimited string
pairs and a matrix where each row has the row label (a
string) as
first field, and the remaining fields constitute the
vector
representing the row label in the vector space of
interest (fields are
tab- or space-delimited).

Output is, for each pair in input, a tab-delimited
line with the two
strings followed by their cosine computed using the
vectors in the
matrix. If one or both the elements in the pair are
not in the matrix,
we return a 0 cosine (but we send a warning to STDERR
if -v option is
specified). We also return 0 as the cosine of anything
with a 0
vector. If a pair is repeated in the input, we repeat
its output.

Usage:

compute_cosines_of_pairs.pl -h

compute_cosines_of_pairs.pl pair-list matrix > cos-
list

NB: The script requires the PDL module.
```

```
Copyright 2009, Marco Baroni

This program is free software. You may copy or
redistribute it under
the same terms as Perl itself.

_USAGE_
}
{
    my $blah = 1;
# this useless block is here because here document
confuses emacs
}

my %opts = ();

getopts('hv',\%opts);

if ($opts{h}) {
    print $usage;
    exit;
}


 # The getopts function takes, as the first
 parameter, a string containing all the possible  #
 arguments of the script (v and h are boolean flags).
 # %opts is a hash table containing the values of the
 arguments: if h is passed, the
 # instructions for the use of the script will be
printed.
```

```
my $target_file = shift;
my $vector_file = shift;

my %target_items = ();
open TARGETS,$target_file
    or die "could not open $target_file";
while (<TARGETS>) {
    chomp;
    s/\r//;
    my ($i,$j) = split "[\t ]+",$_;
    $target_items{$i} = 1;
    $target_items{$j} = 1;
}
close TARGETS;

# $target_file is seen and splitted line by line, and
for each pair of values a flag in the
# hash table $target_items is set to 1 (the keys for
accessing the elements are the values
# themselves, in the variables $i and $j).

if ($opts{v}) {
    print STDERR "target items read in\n";
}

# store normalized vectors
my %vectors = ();

# debug
my $vector_counter = 0;

open VECTORS,$vector_file
    or die "could not open $vector_file";
```

```
while (<VECTORS>) {
    chomp;

    my @F = split "[\t ]+",$_;
    my $item = shift @F;

    if (!(defined($target_items{$item}))) {
     next;
    }


# $vector_file is seen and splitted line by line: for
each line, the function checks whether
# the flag for the corresponding item in $target_items
was set to 1 or not.
# In the second case, the function will pass to the
next iteration.


    my $temp_raw =  pdl @F;
    my $temp_length = sqrt(sum($temp_raw**2));
    if ($temp_length != 0) {
     $vectors{$item} =  $temp_raw / $temp_length;
    }
    else {
     $vectors{$item} = $temp_raw;
    }

    undef($temp_raw);

t# The vector to be summed is saved in $temp_raw,
while the norm of the vector is saved
# in $temp_length: if the norm of the vector is 0,
```

```
$temp_raw will be directly inserted in
# the hash table $vectors ($item will be the key);
otherwise, $temp_raw will be first
# divided by its norm.

    @F = ();

    if ($opts{v}) {
     $vector_counter++;
     print STDERR "$vector_counter in memory\n";
    }
}
close VECTORS;

# If option -v has been specified, the number of
vectors is stored in the variable
# $vector_counter and printed.

%target_items = ();

if ($opts{v}) {
    print STDERR "matrix read and normalized vectors
constructed\n";
}

# If option -v has been specified, a confirmation
message is printed after the reading of the
# matrix and the construction of normalized vectors.

open TARGETS,$target_file
    or die "could not open $target_file the second
time around";
while (<TARGETS>) {
```

```
    chomp;
    s/\r//;
    my ($item1,$item2) = split "[\t ]+",$_;
    my $cosine = 0;

    if ( (!(defined($vectors{$item1}))) ||
      (!(defined($vectors{$item2}))) ) {
     if ($opts{v}) {
         print STDERR "either $item1 or $item2 or both
were not in matrix\n";
     }
    }
    else {
     $cosine = sum($vectors{$item1}*$vectors{$item2});
    }
    printf("%s\t%s\t%.5f\n",$item1,$item2,$cosine);

}
close TARGETS;

# $target_file is seen and splitted line by line.
Then, the function checks whether both the # vectors
for the current items $item1 and $item2 are defined in
the hash table $vectors.
# In the first case, their cosine will be computed and
then printed in the specified format; # in the #
second case, if the option -v has been specified, an
error message will be
# printed.

%vectors = ();

if ($opts{v}) {
```

```
    print STDERR "done\n";
}

# If option -v has been specified, a confirmation
message is printed after the end of the
# process.
```

Python scripts

1) **filter.py**

```
# We used this script to extract -from lists of
results- the top-n words for similarity scores
# with the prototypes.
# The user has to specify the part-of-speech of the
words to be extracted.

f=open(sys.argv[1], "rb")
pos=sys.argv[3]
num=int(sys.argv[4])

# The parameters, respectively the part-of-speech and
of the number of the words to be
# extracted, are assigned to the variables pos and
num.

lista=[]
lista1=[]
stringa=""

for line in f.readlines():
     appoggio=line.split("\t")
     if (str(appoggio[0])[-2:]==pos):
          lista.append(appoggio[2])
          lista1.append(appoggio)
lista.sort(reverse=True)
lista=lista[0:num]

# The file f is seen and splitted line by line. Them,
```

```
if the current word's suffix is
# corresponding to the part-of-speech specified, its
similarity score will be appended to
# lista, while the pair word-score will be appended to
lista1.
# Then the list lista is sorted, and only the top-num
words are preserved.

for elemento in lista:
    i=0
    while i < len(lista1) and
 elemento!=lista1[i][2]:
          i=i+1
    if i < len(lista1):
          stringa=stringa+lista1[i][0]+"
 "+elemento+"\n"

 n=open(sys.argv[2], "wb")
 n.write (stringa)
 n.close()

# The function cycles through lista and lista1 and
prints the highest-scoring words in the
# variable stringa. Then, the variable stringa is
printed in the output file n.
```

2) calculateAccuracy.py

```
# Starting from input files containing the
association scores of the target words with a
# positive or a negative prototype, we used this
script to calculate our accuracy in
# assigning a polarity to the target words.
# The first file passed as a parameter should
contain the association scores with the
# words of the right polarity.

import sys

f1=open(sys.argv[1], "rb")
f2=open(sys.argv[2], "rb")

# f1 and f2 are the input files.

lista=[]
lista1=[]
stringa=""
conta=0

for line in f1.readlines():
    appoggio=line.split("\t")
    lista.append(float(appoggio[1][0:7]))
for line1 in f2.readlines():
    appoggio1=line1.split("\t")
    lista1.append(float(appoggio1[1][0:7]))

# The input files are seen and splitted line by line
and the scores are appended,
# respectively, to lista and lista1.
```

```
def differenza (a, b) :
    return a-b

# The function differenza returns the difference
between two numbers, passed as
# parameters.

for n in map (differenza, lista, lista1):
    if n > 0:
          conta=conta + 1

# The function map cycles through the lists lista
and lista1 and applies the function
# specified as a parameter to every pair of elements
that are in the same position in the
# lists: if the value n returned by differenza is >
0, the variable conta increments by 1.

precisione=(conta*100) / len (lista)
stringa=stringa+"\n"+"Precisione:
"+str(precisione)+" per cento"
n=open(sys.argv[3], "wb")
n.write (stringa)
n.close()

# The accuracy value of the classification is
assigned to stringa and printed in the
# output file.
```

3) **scoreSum.py**

```
# We used this script to sum the similarity scores
between the target words and Turney
# and Littmann's seeds (see chapter 3, paragraph 5,
table 3). We had input files in the
# format  TARGET  SEED  SCORE, and every target word
was present in exactly 14
# tuples, which registered its similarity scores with
each of Turney-Littmann's seed words.

import sys
def extract(in1, in2, out1, out2):

     inNeg = open(in1, 'r')
     inPos = open(in2, 'r')
     outNeg = open(out1, 'w')
     outPos = open(out2, 'w')

# The input files are opened and assigned to the
variables inPos and inNeg.

     scores  = []

     listaP, listaN = {'scores': []}, {'scores': []}
     for line in inNeg.readlines():
          appoggio = line.split("\t")
          if len(appoggio) > 1:
               listaN['scores'].append({'word':
appoggio[0], 'score': appoggio[1]})
     inNeg.close()

     for line in inPos.readlines():
```

```
          appoggio1 = line.split("\t")
          if len(appoggio1) > 1:
               listaP['scores'].append({'word':
appoggio1[0], 'score': appoggio1[1]})
     inPos.close()

# inPos and inNeg are seen and splitted line by line,
and the elements are saved in lista N
# and listaP (dictionaries of dictionaries).
# A if block is needed, because in the input files
there were many empty lines: since the
# method split, if used on an empty line, returns a
list of one element, we include in the
# results only longer lists.

     def extract(data):

# Definition of the function for the sum of the
similarity scores

          j, d = 0, 0
          a, b = [], []

          for i in range(len(data) - 1):
               if data[i]['word'] == data[i -
1]['word']:
                    scores.append({'id': j, 'word':
data[i]['word'], 'score': float(data[i]['score'])})
               else:
                    j = j + 1
                    scores.append({'id': j, 'word':
data[i]['word'], 'score': float(data[i]['score'])})
```

```
# If the word in the list is changed, another ID is
assigned to j, then the ID, the word and
# the score are saved in scores.
# Every position of the scores list will be a
dictionary with id, word and score as keys.


          d = 0
          a, elements = [], []

          for i in range(len(scores) - 1):
               word = scores[i]['word']
               d = d + float(scores[i]['score'])
               if scores[i]['id'] == scores[i +
1]['id']:
                    a.append([word, d])
               else:
                    a.append([word, d])
                    d = 0

# The function cycles through the list scores, and
saves words and similarity scores in the
# new list a.
# The variable d sums the scores for each word and,
when a new ID is encountered, it is
# reset to 0.

          for i in range(0, len(a) - 1):
               if a[i][0] == a[i + 1][0]:
                    pass
               else:
                    elements.append(a[i])
          else:
```

```
                elements.append(a[i])

# The function cycles through the list a and appends
the pair (word, score) to the list
# elements only if the word differs from the next one.
# After the end of the cycle for, also the last pair
(word, score) is appended to elements
# (else block).

          return elements

     positivi = extract(listaP['scores'])

     print "Stampo i positivi: \n"
     for i in positivi:
          tmpStr=str(i[0]) + '\t' + str(i[1])
          outPos.write(tmpStr+'\n')
     outPos.close()

     negativi = extract(listaN['scores'])
     print "\n\nStampo i negativi: \n"
     for i in negativi:
          tmpStr=str(i[0]) + '\t' + str(i[1])
          outNeg.write(tmpStr+'\n')
     outNeg.close()

# The function, which has been defined, is invoked for
the data extraction.
# Finally, the results are printed in the output files
outPos and outNeg.
extract( sys.argv[1], sys.argv[2], sys.argv[3],
sys.argv[4])
```

Compositions and sentiment ratings

The scores range from 1 (very negative) to 5 (very positive). The ratings were collected through a poll on Crowdflower (http://crowdflower.com/), so that each composition has been evaluated by 10 subjects (native English speakers). The triples in red were presented to subjects as the best examples of polarity reversal.

Composition	Average Rating	Composition	Average Rating
to_kill_man	1	to_oppose_tyranny	3,7
to_kill_child	1,4	to_deplore_decision	2,2
to_kill_pain	3,4	to_deplore_method	2,6
to_lose_money	1,5	to_deplore_violence	2,4
to_lose_heart	1,5	to_shame_family	1
to_lose_weight	4,3	to_shame_son	1,1
to_steal_money	1	to_shame_traitor	2,6
to_steal_treasure	1,2	to_evade_law	1,5
to_steal_kiss	3,8	to_evade_police	1,3
to_revoke_permission	1,4	to_evade_temptation	3,9
to_revoke_promise	1,9	to_attack_person	1
to_revoke_punishment	2,7	to_attack_building	1,2
to_hate_man	1,1	to_attack_terrorist	2
to_hate_brother	1,5	to_demolish_building	2,3
to_hate_injustice	3,6	to_demolish_reputation	1,3
to_delay_train	2	to_demolish_prejudice	4,1
to_delay_meeting	2,1	to_blackmail_wife	1
to_delay_catastrophe	3	to_blackmail_politician	1,2
to_disapprove_decisio	2,3	to_blackmail_criminal	1,5

n			
to_disapprove_obedience	1,8	to_frustrate_child	1,4
to_disapprove_violence	4,4	to_frustrate_expectation	1,7
to_criticize_decision	2,1	to_frustrate_conspiracy	1,9
to_criticize_behaviour	1,8	to_disappoint_friend	1,5
to_criticize_conformism	2,6	to_disappoint_family	1,1
to_oppose_government	1,6	to_disappoint_gangster	2
to_oppose_decision	2,4	to_worship_God	5
to_abandon_dog	1,1	to_worship_friend	3,9
to_abandon_child	1	to_worship_devil	1
to_abandon_doubt	3,2	to_sanctify_man	3,8
to_sabotage_plant	1,4	to_sanctify_pope	4,1
to_sabotage_operation	1,5	to_sanctify_criminal	2
to_sabotage_conspiracy	2,6	to_legitimate_act	3,9
to_censure_idea	2,2	to_legitimate_marriage	4,6
to_censure_opinion	2,2	to_legitimate_crime	1,6
to_censure_insult	2,5	to_profess_love	4,9
to_remove_agent	2,4	to_profess_faith	3,7
to_remove_politician	2,4	to_profess_communism	2,3
to_remove_traitor	3,5	to_admire_person	4,2
to_obtain_citizenship	4,1	to_admire_friend	4,8
to_obtain_studentship	4,2	to_admire_criminal	1,3
to_obtain_revenge	2,1	to_encourage_friend	4,9
to_hail_friend	3,8	to_encourage_team	4,9
to_hail_king	4	to_encourage_crime	1,6
to_hail_tyrant	2	to_vote_politician	3,5
to_help_friend	4,9	to_vote_party	4,1
to_help_kin	4,9	to_vote_racist	1,6

to_help_criminal	1,1	to_simplify_exercise	3,9
to_flirt_idea	3	to_simplify_negotiation	3,8
to_flirt_man	4,1	to_simplify_evasion	2,6
to_flirt_disaster	2,6	to_inspire_hope	4,9
to_marry_man	4,7	to_inspire_confidence	5
to_marry_woman	4,9	to_inspire_violence	1,1
to_marry_criminal	1,4	to_desire_freedom	4,3
to_play_friend	3,9	to_desire_happiness	4,7
to_play_team	4,2	to_desire_death	1,1
to_play_fire	1,6	to_excite_imagination	4,9
to_revere_friend	3,6	to_excite_happiness	4,8
to_revere_king	3,6	to_excite_insurrection	2,1
to_revere_devil	1,8	to_embrace_cause	4,1
to_approve_decision	4,4	to_embrace_ideal	4
to_approve_obedience	3,8	to_embrace_socialism	3,1
to_approve_violence	1,1	to_grow_importance	4,5
to_love_friend	4,9	to_grow_wisdom	4,7
to_love_enemy	4,8	to_grow_malice	1,6
to_love_sin	1,4	to_excel_art	4,1
to_pay_debt	4,4	to_excel_game	4
to_pay_money	3,3	to_excel_devil	1,9
to_pay_price	2,7	to_promise_prize	4,6
		to_promise_reward	4,4
		to_promise_illusion	3,2

Bibliography

- A. Almuhareb, M. Poesio, *Attribute-based and value-based clustering: an evaluation,* in *Proceedings of the EMNLP*, Barcelona (SPA), 2004;
- F. Baccianella, A. Esuli, F. Sebastiani, *SentiWordNet 3.0: an enhanced lexical resource for Sentiment Analysis and Opinion Mining,* in *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC 2010),* Valletta (MT), 2010;
- M. Baroni *et al.*, *Strudel: a corpus-based semantic model based on properties and types*, in *Cognitive Science*, vol. 34, n. 10, 2010;
- M. Baroni, A. Lenci, *Concepts and properties in word spaces*, in *Rivista di Linguistica*, vol. 20, n. 1, 2008;
- M. Baroni and A. Lenci, *Distributional Memory: a general framework for corpus-based semantics,* in *Computational Linguistics,* vol. 36, n. 4, 2010;
- M. Baroni, R. Zamparelli, *Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space*, in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, Association for Computational Linguistics, East Stroudsburg (Pennsylvania), 2010;
- L. Barsalou, *Language and simulation in conceptual processing*, in *Symbols, embodiment and meaning*, edited by M. De Vega *et al.*, Oxford University Press, Oxford, 2008;
- L. Barsalou, *Perceptual symbol systems,* in *Behavioral and Brain Sciences*, no. 22, 1999;
- D. Blei *et al.*, *Latent Dirichlet Allocation*, in *Journal of Machine Learning Research,* vol. 3, no. 4, 2003;
- J. Blitzer, M. Dredze, F. Pereira, *Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification,* in *Proceeding of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, Prague (CZE), 2007;
- C. Brockmann, M. Lapata, *Evaluating and combining approaches to selectional preference acquisition*, in *Proceedings of the European Chapter of the Association for Computational Linguistics,* Association for Computational Linguistics, 2003;
- C. Cardie, Y. Choi, *Learning with compositional semantics as structural inference for subsentential Sentiment Analysis*, in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008*), Waikiki (Hawaii),

2008;

- W. G. Charles, *Contextual correlates of meaning,* in *Applied Psycholinguistics*, vol. 21, no. 4, Cambridge University Press, 2000;
- N. Chomsky, *Aspects of the theory of syntax,* MIT Press, Cambridge (MA), 1965;
- K. Church, P. Hanks, *Word association norms, mutual information, and lexicography*, in *Proceedings of the 27th Annual Conference of the Association of Computational Linguistics*, Vancouver (British Columbia), 1989;
- S. Clark, *Vector Space Models of lexical meaning*, in *Handbook of Contemporary Semantics*, edited by S. Lappin, C. Fox, Wiley-Blackwell, 2012;
- J. G. Conrad, F. Schilder, *Opinion mining in legal blogs,* in *Proceedings of the International Conference on Artificial Intelligence and Law (ICAIL),* New York, 2007;
- G. Cree, K. McRae, *Analyzing the factors underlying the structure and computation of the meaning of* chipmunk, cherry, chisel, cheese *and* cello *(and many other such concrete nouns)*, in *Journal of Experimental Psychology*, vol. 132, no. 2, 2003;
- W. Croft, A. Cruse, *Cognitive linguistics*, Cambridge Textbooks in Linguistics, Cambridge University Press, 2004,
- A. Cruse, *Meaning in language: an introduction to semantics and pragmatics*, Oxford Textbooks in Linguistics, Oxford University Press, 2004,
- A. Das, S. Bandyopadhyay, *Towards the Global SentiWordNet*, *Proceedings of the 24th Pacific Asia Conference on Language Information and Computation 2010*, Tohoku University (Japan), 2010, pp. 799-808;
- S. Das, M. Chen, *Yahoo! For Amazon: extracting market sentiment from stock message boards*, in *Proceedings of the 8th Asia Pacific Finance Association Annual Conference*, Bangkok, 2001;
- A. Das, B. Gambäck, *Sentimantics: Conceptual Spaces for Lexical Sentiment Polarity Representation with Contextuality,* in *Proceedings of the 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, Jeju (KOR), 2012;
- K. Dave *et al.*, *Mining the peanut gallery: opinion extraction and semantic classification of product reviews*, in *Proceedings of the 12th International Conference on the World Wide Web*, Budapest, 2003;

- S. Deerwester *et al.*, *Indexing by Latent Semantic Analysis*, in *Journal of the American Society for Information Science*, vol. 41, no. 6, 1990;
- K. Erk *et al.*, *A flexible, corpus-driven model of regular and inverse selectional preferences,* in *Journal of Computational Linguistics,* MIT Press, Cambridge (Massachusetts), vol. 36, no. 4, 2010;
- K. Erk, *A simple, similarity-based model for selectional preferences*, in *Proceedings of the Association for Computational Linguistics,* Association for Computational Linguistics, 2007;
- K. Erk, S. Padó, *A structured vector space model for word meaning in context*, in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP – 08),* Honolulu (Hawaii), 2008;
- K. Erk, S. Padó, *Exemplar-based models for word meaning in context*, in *Proceedings of 2010 Conference of the Association for Computational Linguistics,* Association for Computational Linguistics, Uppsala (SWE), 2010;
- A. Esuli, *Automatic generation of Lexical Resources for Opinion Mining: models, algorithms and applications,* PhD Thesis, PhD School on Information Engineering "Leonardo da Vinci", University of Pisa, 2008;
- A. Esuli, F. Sebastiani, *Determining term subjectivity and term orientation for opinion mining*, in *Proceedings of EACL-06*, *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, 2006;
- A. Esuli, F. Sebastiani, *SentiWordNet: a publicly available lexical resource for Opinion Mining*, in *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC 2006),* Genova, 2006;
- C. Fessbaum, *Wordnet: an electronic lexical database*, MIT Press, Cambridge, 1998;
- C. Fellbaum, *WordNet and wordnets,* in K. Brown *et alii, Encyclopedia of Language and Linguistics*, Elsevier, Oxford, 2005;
- J. Firth, *Papers in Linguistics 1934-1951*, Oxford University Press, London, 1957;
- J. Fodor, Z. Pylyshyn, *Connectionism and cognitive architecture: a critical analysis,* in *Cognition*, vol. 28, Elsevier, 1988;
- F. Foltz *et al.*, *The intelligent essay assessor: applications to educational technology*, in *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, vol. 1, no. 2, 1999;

- G. Furnas *et al.*, *Statistical semantics: analysis of the potential performance of keyword information systems*, Bell System Technical Journal, vol. 62 , no. 6, 1983;
- D. Gentner, *Structure-mapping: a theoretical framework for analogy*, in *Cognitive Science*, vol. 7, no. 2, 1983;
- E. Giesbrecht, *In search of semantic compositionality in Vector Spaces*, in *Proceedings of International Conference on Computational Science*, Moscow (RUS), 2009;
- E. Giesbrecht, *Towards a matrix-based distributional model of meaning,* in *Proceedings of Human Language Technologies: the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics – Student Research Workshop,* Association for Computational Linguistics, 2010;
- A. Glenberg, S. Mehta, *Constraint on covariation: it's not meaning*, in *Rivista di Linguistica*, vol. 20, no. 1, 2008;
- A. Glenberg, M. Robertson, *Symbol grounding and meaning: a comparison of high-dimensional and embodied theories of meaning*, in *Journal of Memory and Language*, no. 43, Elsevier, 2000;
- G. Golub, C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996;
- E. Grefenstette *et al.*, *Concrete sentence spaces for compositional distributional models of meaning,* in *Proceedings of the 9th International Conference on Computational Semantics,* 2011;
- E. Guevara, *A regression model of adjective-noun compositionality in Distributional Semantics,* in *Proceedings of the 2010 Workshop on Geometrical Models of Natural Language Semantics*, Association for Computational Linguistics, Uppsala (SWE), 2010;
- E. Guevara, *Computing semantic compositionality in distributional semantics,* in *Proceedings of the 9th International Conference on Computational Semantics,* Association for Computational Linguistics, Stroudsburg (Pennsylvania), 2011;
- Z. Harris, *Distributional structure*, in *Papers in structural and transformational Linguistics*, Formal Linguistics Series, vol. 1, Humanities Press, New York, 1970;
- Z. Harris, *Methods in structural linguistics*, University of Chicago Press, Chicago, 1951;
- V. Hatzivassiloglou, K. McKeown, *Predicting the semantic orientation of adjectives,* in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 18th*

Conference of the European Chapter of the ACL, Association for Computational Linguistics, New Brunswick (NJ), 1997;

- M. Hu, B. Liu, *Mining and summarizing customer reviews*, in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP),* Vancouver, 2005;
- H. Jang, J. Mostow, *Inferring selectional preferences from part-of-speech N-grams,* in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics,* Association for Computational Linguistics, Stroudsburg (Pennsylvania), 2012;
- K. S. Jones, *A statistical interpretation of term specificity and its application in retrieval*, Journal of Application, Emerald, vol. 28, no. 1, 1972;
- M. N. Jones, D. J. K. Mewhort, *Representing word meaning and order information in a composite holographic lexicon*, in *Psychological Review,* vol. 114, 2007;
- A. Kale *et al.*, *Modeling trust and influence in the blogosphere using link polarity*, in *Proceedings of the International Conference on Weblogs and Social Media (ICWSM),* Boulder (Colorado), 2007;
- J. Kamps *et alii*, *Using WordNet to measure semantic orientation of adjectives*, in *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC-04)*, vol. IV, Lisbon, 2004;
- G. Karpys, Y. Zhao, *Evaluation of hierachical clustering algorithms for documents datasets*, in *Proceedings of the 11th International Conference on Information and Knowledge Management,* McLean (Virginia), 2002;
- J. J. Katz, J. Fodor, *The structure of semantic theory*, in *The structure of language*, Prentice-Hall, 1964; Y. Wilks, *Preference semantics*, in *Formal semantics of Natural Language*, Cambridge University Press, Cambridge (UK), 1975;
- C. Kelly *et al.*, *Acquiring human-like feature-based conceptual representations from corpora*, in *Proceedings of the NAACL HLT 2010 - 1st Workshop on Computation Neurolinguistics*, Los Angeles, 2010;
- C. Kelly *et al.*, *Semi-supervised learning for automatic conceptual property extraction*, in *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2012)*, Montreal, 2012;

- A. Kennedy, D. Inkpen, *Sentiment Classification of Movie Reviews Using Contextual Valence Shifters*, Journal of Computational Intelligence, vol. 22, n. 2, 2006;
- A. Kilgarriff, *I don't believe in word senses*, Computers and the Humanities, Springer, vol. 31, 1997;
- A. Kilgarriff *et al.*, *The Sketch Engine*, in *Proceedings of Euralex*, Lorient (FRA), 2004;
- S. Kim *et al.*, *Automatically assessing review helpfulness*, in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006;
- S. Kim, E. Hovy, *Determining the sentiment of opinions*, in *Proceedings of the 20th International Conference on Computational Linguistics,* Geneva (SUI), 2004;
- W. Kintsch, *Predication,* in *Cognitive Science,* vol. 25, no. 2, 2001;
- T. Landauer, *On the computational basis of learning and cognition: Arguments from LSA*, in *The Psychology of Learning and Motivation*, edited by B.H. Ross, Elsevier, 2002;
- T. Landauer *et al.* (edited by), *The handbook of Latent Semantic Analysis*, Lawrence Erlbaum, Mahwah (New Jersey), 2007;
- T. Landauer, S. Dumais, *A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge*, in *Psychological Review*, vol. 104, no. 2, 1997;
- D. Lee, O. Jeong, S. Lee, *Opinion mining of customer feedback data on the Web,* in *Proceedings of the 2nd international conference on Ubiquitous information management and communication ICUIMC,* Association for Computing Machinery, New York, 2008;
- A. Lenci, *Distributional semantics in linguistic and cognitive research. A foreword*, in *Rivista di Linguistica*, vol. 20, no. 1, 2008;
- D. Lin, *Automatic retrieval and clustering of similar words,* in *Proceedings of the 17th International Conference on Computational Linguistics,* Association for Computational Linguistics, 1998;
- D. Lin, P. Pantel, *DIRT – Discovery of Inference Rules from Text,* in *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining,* San Francisco (California), 2001;
- D. Lin, P. Pantel, *Document clustering with committees,* in *Proceedings of the 25th Annual International ACM SIGIR Conference,* Tampere (Finland), 2002;
- B. Liu, *Sentiment Analysis and Opinion Mining*, in *Synthesis Lectures*

on Human Language Technologies, edited by G. Hirst, Morgan and Claypool Publishers, 2012;

- H. Liu, P. Singh, *ConceptNet: a practical commonsense reasoning toolkit*, in *BT Technology Journal*, vol. 22, no. 4, 2004;
- K. Lund, C. Burgess, *Producing high-dimensional semantic spaces from lexical co-occurrence*, in *Behavior Research Methods, Instruments and Computers*, vol. 28, no. 2, 1996;
- A. Maas *et al.*, *Learning word vectors for Sentiment Analysis*, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, Stroudsburg (Pennsylvania), 2011;
- C. Manning *et al.*, I*ntroduction to Information Retrieval*, Cambridge University Press, Cambridge, 2008;
- C. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing,* MIT Press, Cambridge (Massachusetts), 1999;
- K. McRae *et al.*, *Semantic feature production norms for a large set of living and nonliving things,* in *Behavioral Research Methods, Instruments and Computers,* vol. 37, 2005;
- Y. Mejova, *Sentiment analysis: an overview*, Comprehensive exam paper, University of Iowa, 2009;
- G. A. Miller, *Wordnet: a lexical database for English*, Communication of the Association for Computing Machinery, vol. 38, n. 11, 1995;
- G. A. Miller, W. G. Charles, *Contextual correlates of semantic similarity*, in *Language and Cognitive Processes*, vol. 6, no. 1, Taylor & Francis, 1991;
- J. Mitchell, M. Lapata, *Composition in distributional models of semantics,* in *Cognitive Science,* vol. 34, no. 8, 2010;
- J. Mitchell, M. Lapata, *Vector-based models of semantic composition*, in *Proceedings of the Association of Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Columbus (Ohio), 2008;
- S. Mohammad, P. D. Turney, *Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon*, In *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, Los Angeles (California), 2010;
- S. Mohammad, P. D. Turney, *Crowdsourcing a Word-Emotion*

Association Lexicon, to appear in *Computational Intelligence*, Wiley Blackwell Publishing Ltd;

- C. Morris, *Foundations of a theory of signs*, in *International Encyclopedia of Unified Science*, vol. 1, University of Chicago Press, Chicago, 1938;
- A. Moschitti, S. Quarteroni, *Kernels on linguistic structures for answer extraction*, in *Proceedings of the Association of Computational Linguistics: Human Language Technologies,* Association for Computational Linguistics, Columbus (Ohio), 2008;
- G. L. Murphy, *The big book of concepts,* MIT Press, Cambridge (Massachusetts), 2002;
- F. A. Nielsen, *A new ANEW: Evaluation of a word list for sentiment analysis in microblogs*, *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages 718*, in *CEUR Workshop Proceedings*, 2011;
- C. E. Osgood, *et al.*, *The measurement of meaning*, University of Illinois Press, Urbana (Illinois), 1957;
- S. Padó, M. Lapata, *Dependency-based construction of semantic space models*, in *Computational Linguistics,* vol. 33, no. 2, 2007;
- B. Pang, L. Lee, *A sentimental education : Sentiment Analysis using subjectivity summarization based on minimum cuts, i*n *Proceedings of 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, Barcelona (SPA), 2004, pp. 271-278;
- B. Pang, L. Lee, *Opinion mining and sentiment analysis*, in *Foundations and trends in Information Retrieval,* vol. 2, n. 1-2, 2008;
- B. Pang, L. Lee, *Seeing stars: exploiting class relationship for sentiment categorization with respect to rating scales*, in *Proceedings of 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP 2005),* Vancouver, 2005;
- B. Pang, L. Lee, S. Vaithyanathan, *Thumbs up? Sentiment Classification using machine learning techniques*, *Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, Philadelphia, 2002;
- P. Pantel, M. Pennacchiotti, *Espresso: leveraging generic patterns for automatically harvesting semantic relations,* in *Proceedings of COLING-ACL,* Sydney, 2006;
- P. Pantel, P. Turney, *From frequency to meaning: vector space models for semantics*, Journal of Artificial Intelligence Research, no. 37, 2010;

- B. H. Partee, A. Ter Meulen, R. E. Wall, *Mathematical methods in linguistics*, Kluwer, Dordrecht (NL), 1990;
- T. Pedersen, *Unsupervised corpus-based methods for Word Sense Disambiguation,* in E. Agirre, P. Edmonds, *Word Sense Disambiguation: Algorithms and Applications*, Springer, 2006;
- T. A. Plate, *Holographic reduced representations: convolution algebra for compositional distributed representations,* in *Proceedings of the 12th International Joint Conference on Artificial Intelligence,* Sydney (AUS), 1991
- L. Polanyi, A. Zaenen, *Contextual valence shifters*, in *Computing attitude and affect in text: theory and applications*, edited by J. Wiebe, Springer, Dordrecht, 2004;
- A. Popescu, O. Etzioni, *Extracting product features and opinions from reviews*, in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP),* Vancouver, 2005;
- R. Quirk, *A comprehensive grammar of the English language*, Longman, London, 1985;
- P. Resnik, *Selectional constraints: an information-theoretic model and its computational realization*, in *Cognition*, vol. 61, 1995;
- K. Rothenhuäsler, H. Schütze, *Unsupervised classification with dependency based word spaces,* in *Proceedings of the EACL GEMS Workshop,* Athens (GRE), 2009;
- S. Rudolph, E. Giesbrecht, *Compositional matrix-space models of language,* in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Uppsala (SWE), 2010
- M. Sahlgren, *An introduction to Random Indexing*, in *Proceedings of the Methods and Applications of Semantic Indexing Workshop,* at the *9th International Conference on Terminology and Knowledge Engineering*, Copenaghen (DEN), 2005;
- M. Sahlgren, *The distributional hypothesis,* in *Rivista di Linguistica*, vol. 20, no. 1, 2008;
- M. Sahlgren, *The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces,* PhD Thesis, Department of Linguistics, Stockholm University, 2006;

- G. Salton *et al.*, *A vector space model for automatic indexing*, in *Communications of the ACM (Association for Computing Machinery)*, vol. 18, no. 11, 1975;
- G. Salton, C. Buckley, *Term-weighting approaches in automatic text retrieval*, in *Information Processing and Management*, vol. 24, no. 5, 1988;
- H. Schütze, *Automatic word sense discrimination,* in *Journal of Computational Linguistics,* MIT Press, Cambridge (Massachusetts), vol. 24, no. 1, 1998
- H. Schütze, J. Pederson, *Information retrieval based on word senses*, in *Proceedings of ACM Special Interest Group on Information Retrieval (SIGIR)*, 1995;
- F. Sebastiani, *Machine learning in automated text categorization*, in *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, 2002;
- C. Shannon, *A mathematical theory of communication*, in *Bell System Technical Journal*, University of Illinois Press, vol. 27, 1948;
- A. Singhal *et al.*, *Document length normalization*, in *Information processing and management*, vol. 32, no. 5, 1996;
- P. Smolensky, *Tensor product variable binding and the representation of symbolic structures in connectionist systems*, in *Journal of Artificial Intelligence,* vol. 46, 1990;
- R. Socher *et al.*, *Semantic Compositionality through recursive matrix-vector spaces,* in *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP 2012*), Jeju (KOR), 2012;
- R. Socher *et al.*, *Semi-Supervised autoencoders for predicting Sentiment Distributions,* in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh (SCO), 2011;
- C. Strapparava, A. Valitutti, *Wordnet-affect: an affective extension of wordnet, i*n *Proceedings of the* 4^{th} *International Conference on Language Resources and Evaluation*, Lisbon (POR), 2004, pp. 1083-1086;
- P. Subasic, A. Huettner, *Affect analysis of text using fuzzy semantic typing*, Institut of Electric and Electronics and Engineering – Finland Section, vol. 9, n. 4, pp. 483-496, 2001;

- M. Taboada *et alii*, *Lexicon-based methods for Sentiment analysis*, Computational Linguistics, MIT Press Cambridge, Boston, Vol. 37, n. 2, 2011, pp. 267-307;
- P. N. Tan *et al., Introduction to Data Mining*, Pearson Addison Weasley, Boston (Massachusetts), 2006;
- M. Thomas, B. Pang, L. Lee, *Get out the vote: determining support or opposition from Congressional Floor-debate transcripts*, in *Proceedings of 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, 2006;
- P. D. Turney, *A uniform approach to analogies, synonyms, antonyms and associations,* in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester (UK), 2008;
- P. D. Turney *et al.*, *Combining indipendent modules to solve multiple-choice synonym and analogy problems*, in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, Borovets (Bulgaria), 2003;
- P. D. Turney, *Mining the web for synonyms: PMI-IR versus LSA on TOEFL*, in *Proceedings of the 12th European Conference on Machine Learning (ECML 2001)*, Freiburg (GER), 2001;
- P. D. Turney, *Similarity of semantic relations,* in *Computational Linguistics,* vol. 32, no. 3, 2006;
- P. D. Turney, *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*, in *Proceedings of the 40 th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, 2002;
- P. Turney, M. Littman, *Measuring praise and criticism: inference of semantic orientation from association*, ACM Transactions on Information Systems, vol. 21, n. 4, 2003;
- G. Vigliocco *et al.*, *Toward a theory of semantic representation*, in *Language and* Cognition, vol. 1, no. 2, 2009;
- A. Warriner *et al.*, *Norms of valence, arousal and dominance for 13915 English lemmas,* to appear in *Behaviour Research Methods*;
- D. Widdows, *Geometry and Meaning,* CSLI Publications, Stanford, 2004;
- D. Widdows, *Semantic Vector Products: some initial investigations*, in *Second AAAI Symposium on Quantum Interaction,* Oxford, 2008;

- J. M. Wiebe *et al.*, *Learning subjective language*, Computational Linguistics, MIT Press Journals, vol. 30, n. 3, 2004;
- J. Wiebe *et al.*, *Annotating expressions of opinions and emotions in language*, *Language Resources and Evaluation*, vol. 39 (2-3), 2005, pp. 165-210;
- T. Wilson *et al.* , *Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis*, in *Proceedings of HLT-EMNLP,* 2005;
- D. Yarowsky, *Unsupervised word sense disambiguation rivaling supervised methods,* in *Proceedings of the 33rd Annual Meeting of the Association of Computational Linguistics,* Association for Computational Linguistics, Cambridge (Massachusetts), 1995
- A. Yessenalina, C. Cardie, *Compositional matrix-space models for Sentiment Analysis,* in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011*), Edinburgh (SCO), 2011;
- H. Yu, V. Hatzivassiloglou, *Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences*, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP),* Sapporo (JAP), 2003;
- D. Yuret, M. Yatbaz, *The noisy channel model for unsupervised word sense disambiguation*, in *Computational Linguistics,* vol. 36, no. 1, 2010;
- F. M. Zanzotto *et al.*, *Estimating linear models for Compositional Distributional Semantics*, in *Proceedings of the 23rd International Conference on Computational Linguistics,* Association for Computational Linguistics, Bejing (CHN), 2010.

Webliography

- http://clic.cimec.unitn.it/dm/

The website of Distributional Memory, a framework for Distributional Semantics;

- http://conceptnet5.media.mit.edu/

The website of the project ConceptNet: ConceptNet is a "commonsense network", built from nodes representing concepts and labeled relationships between them;

- http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

The website of the Multi-Domain Sentiment dataset, a dataset consisting of product reviews from four different product types (books, electronics, DVDs and kitchen appliances), with 1000 positive and 1000 negative reviews for each of these categories

- http://www.cs.pitt.edu/mpqa/index.html.

A list of subjectivity clues, downloadable from the Opinion Finder's website;

- http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

The Hu-Liu list, a list containing positive and negative opinion words or sentiment words for the English language (around 6800 words) compiled over many years;

- http://www.cs.cornell.edu/home/llee/data/convote.html

The website of the Congressional Floor-Debate corpus, a congressional-speech corpus, including a total of 3857 speech segments transcribed from 53 different debates;

- http://www.cs.cornell.edu/people/pabo/movie-review-data/

The website of the Cornell movie-review dataset. From the website, it is possible to download collections of movie-review with labels indicating the overall sentiment polarity (positive or negative) or subjective rating (the number of stars assigned) of the documents, for a total of 1000 positive and 1000 negative reviews, or collections of sentences labeled with respect to their subjectivity status, for a total of 5000 subjective and 5000 objective processed sentences;

- http://gate.ac.uk/download/

The website of the MPQA opinion corpus, a corpus containing 535 articles from a wide variety of news sources manually annotated for opinions and other private states;

- http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

Description and links for the download of the AFINN list, a lexicon projected for sentiment analysis in microblogs;

- http://www.wjh.harvard.edu/~inquirer/Home.html

The website provides entry-points to resources associated with the General Inquirer. It's possible to find here lists of manually-classified terms with various kinds of markers (semantic orientation, cognitive orientation, mood of the speaker etc.);

- http://sentiwordnet.isti.cnr.it/

SentiWordNet is a lexical resource for Opinion Mining, in which three sentiment scores -positivity, negativity, neutrality- are assigned to each synset of WordNet;

- http://www.sketchengine.co.uk/

The website of the Corpus Query System Sketch Engine.

- http://www.umiacs.umd.edu/~saif/WebPages/ResearchInterests.html

The personal page of Mohammed Saif, researcher at the Institute for Information Technology, National Research Council Canada (NRC);

- www.wikipedia.it

Webpage of the Wikipedia project;

- www.wordreference.com

Webpage of the online dictionary wordreference.com;

- http://wndomains.fbk.eu/index.html

WordNet Affect is an extension of WordNet Domains, a lexical resource created in a semi-automatic way by augmenting WordNet with domain labels; the extension WordNet Affect consisted in the addition of another set of synsets representing affective concepts.