UNIVERSITÀ DI PISA

Corso di Laurea in

Informatica Umanistica

RELAZIONE

Dimensionality reduction on syntax-based distributional semantics models: the case of crosslingual and multilingual distributional memories for German

Candidato: *Laura Aina*

Relatore: *Prof. Alessandro Lenci*

Correlatori: *Prof. Sebastian Padò,*
Prof.ssa Maria Simi

Anno Accademico 2013-2014

The project that is described as content of this thesis has been conducted by the candidate during an Erasmus traineeship at the Institüt für Maschinelle Sprachverarbeitung (IMS) of the University of Stuttgart, from the 1st September to the 1st December 2014.

Summary

Introduction

In the field of distributional semantics, distributional memories (DMs) are a useful and robust syntax-based method of word representation. Assuming distributional hypothesis and thus contexts of co-occurrence being highly representative of a word meaning, they consist in sets of tuples <*word*, *link*, *word*> organized in third-order tensors (Baroni and Lenci, 2010). From these structures different kinds of matrices can then be built depending on the type of semantic information that needs to be extracted for a given task.

Translating distributional memories has resulted in being a high quality method of building new ones, only using the DM of a source language and a translation lexicon for the target language (Padò and Utt, 2012). This way, it is possible to exploit the high number and the good performances of the resources available for English, that are often missing for other languages.

Nevertheless, the DM obtained by translation for German (Padò and Utt, 2012) shows a disadvantage which is related to its size, as entries are much more than in the English one, while words are significantly less. Visualizing a DM as a graph, a crosslingual DM results in a much denser graph with low amount of nodes (words) and a high amount of edges (between two words and labeled with the type of link).

On one hand, the use of manually compiled translation lexicons reduces the level of lexical coverage, comparing it to the one obtained instead taking advantage of a large corpus. On the other, lexical ambiguity, both in the source and in the target language, considerably increases the number of edges. As a consequence of these two factors, the dimensions of DMs constructed by translation happen to be particularly large.

Therefore, efficient methodologies of size reduction of DMs obtained via crosslingual methods would allow a better exploit of these resources, if they can keep the semantic properties of the model without loss of the most relevant information.

During the project, whose aim has been to address this latter issue, the approach that has been considered was dimensionality reduction, that is a transformation over the structure of a semantic space that is supposed to reduce the size of the model and to discover latent information by generalizing over the model (Van den Cruyus, 2010). Its potentiality has been investigated, looking at its practicality, efficiency, quality and the relationship of the resulting models with the original one.

In practice, methods of matrix factorization have been applied on the structure obtained by the matricization of the DM, previously reduced by filtering a fixed number of most relevant link-word pairs. Two kinds of transformations have been considered: singular value decomposition (SVD) and non-negative matrix factorization (NMF). With these methods, the space can be reduced to 500 dimensions.

This kind of reduction has been implemented on the DM built via translation for the German language, DM.Xl (Padò and Utt, 2012). The data resulting from the reduction have then been evaluated by comparing them with the already available German DM resources: an original DM built via a crosslingual method, a DM built via traditional method (DM.De; Padò and Utt, 2012), a DM built via multilingual method (DM.MULTI; Padò and Utt, 2012), which combines the previous two.

The parameter for the evaluation has been the performances of the models in the task of word similarity prediction, that is observing how much relatedness values assigned by the model correspond to those assigned by native speakers.

Moreover, if a DM obtained by traditional method (monolingual) is available for the target language, merging it with one built by translation from English enables to exploit both English and the target language resources and to have the complementary properties of the two DMs in a single model. Therefore, it would then be possible to benefit of the good quality of the crosslingual DM and the higher coverage of the monolingual one. A multilingual DM has been obtained for German combining the resulting similarities (DM.MULTI; Padò and Utt, 2012).

During the project, another method that makes use of dimensionality reduction has been tested: the two available German DMs are concatenated and then transformations are applied on the resulting matrix. This way, the size of the merged DMs, already reduced with the selection of the top link-word pairs, is then further kept lower. The concatenation of the models can be applied in two different moments of the processing, so two different models have been built and the effects on the size and the balance between the entries belonging to each DMs have been observed and compared. Both the effects of SVD and NMF have been tested.

The effects of dimensionality reduction has then been observed in these two experiments, in order to discover whether this approach could be a manageable and high-quality method of getting the advantages of a low data size achieving an equal or better capacity of describing semantic phenomena.

1. Distributional memories

Distributional memories (DMs) are a general framework for building a model than contains distributional information extracted from a corpus, in the form of weighted word-link-word tuples arranged in a third-order tensor (Baroni and Lenci, 2010).

Typically, in corpus-based semantics an ad-hoc model is built depending on the different kind of semantic information that needs to be collected. DMs approach allows instead to generate different types of matrices from the tensor, so that different semantic tasks, such as word-similarity judgments, discovering synonyms, concept categorization, selectional preferences and relations between word pairs, can be addressed by the very same model.

1.1 Distributional semantics models and distributional memories

Corpus-based semantic models of semantic representation, also known as distributional semantics models (DSMs), all rely on distributional hypothesis (Harris, 1954; Miller and Charles, 1991): the degree of semantic similarity between two words can be modeled as a function of the degree of overlap among their linguistic contexts. In other words, two terms are as similar to each other as much as they share contexts where they are used.

Co-occurrence values stored in a matrix that has as rows words, or any other target linguistic elements, and as columns the contexts feature is called a semantic space and describes the word distribution within contexts and so its meaning according to the distributional hypothesis. Therefore, terms can be represented as high-dimensional vectors (row vectors in the matrix), where the dimensions correspond to context features (Turney and Pantel, 2010).

Semantic relatedness, or attributional similarity, between two words a and b, $sim\ (a, b)$, depends on the degree of correspondence between the properties of a and b (Turney and Pantel, 2010). In other words, two words are semantically related to the degree that they share attributes. Examples are synonyms (*bank* and *trust company*), meronyms (*car* and *wheel*), antonyms (*hot* and *cold*), and words that are functionally related or frequently associated (*pencil* and *paper*).

Assuming distributional hypothesis, semantic relatedness, as it is supposed to come with a similar distribution over contexts of the target words, can consequently be measured by comparing the respective word vectors in a semantic space, relying on the fact that these vectors capture the semantic content of a word.

Though all the DSMs are based on these assumptions, different approaches have been proposed depending on the aspects of meaning they are designed to model, like attributional or relational similarity. While the former is involved in taxonomic semantic relations (e.g. synonymy, hyponymy), the latter is the property shared by pairs of words linked by similar semantic relations (e.g. hypernymy).

DMs framework stems from the argument that the “one semantic task, one distributional model” approach has various limits. As a matter of fact, though these representations are supposed to model on a large scale linguistic information acquisition and use, human semantic competence, which typically in cognitive science is related to a single semantic memory (Murphy, 2002; Rogers and McClelland, 2004), has a multipurpose nature that DSMs of this kind lack. Thus, a generalized framework for distributional semantics model has been introduced on the assumption that it is the choice of representing co-occurrences statistics as matrices that entails the lack of generalization. The standard view indeed models semantic properties in terms of two-way structures, that is matrices coupling target elements and context (Padò and Lapata, 2007). DMs tensor structure allows instead the generation of different matrices that correspond to different “views” of the same data, extracted once and for all from a corpus.

1.2 The distributional memory framework

DM represent distributional data with weighted tuple structures (Baroni and Lenci, 2010).

Let O_1 and O_2 be two sets of objects, and $R \subseteq O_1 \times O_2$ a set of relations between these objects. A triple $< o_1, r, o_2 >$ expresses the fact that o_1 is linked to o_2 through the relation r. Weighted distributional tuples, included in a DM, encodes distributional facts in terms of typed co-occurrence relations among words. Let W_1 and W_2 be sets of strings representing content words, and L a set of strings representing syntagmatic co-occurrence links between words in a text. $T \subseteq W_1 \times L \times W_2$ is a set of corpus-derived tuples $t = < w_1, l, w_2 >$, such that co-occurs with and l represents the type of this co-occurrence relation. For instance, the tuple <*marine*, *use*, *bomb*> encodes the piece of distributional information that *marine* co-occurs with *bomb* in the corpus by the syntagmatic link *use*.

Each tuple t has a weight, a real-valued score v_t, assigned by a scoring function $\sigma: W_1 \times L \times W_2 \rightarrow \mathbb{R}$. A weighted tuple structure consists of the set T_W of weighted distributional tuples $t_w = < t, v_t >$ for all $t \in T$ and $\sigma(t) = v_t$.

It is assumed that $W_1 = W_2$ and an inverse link constraint is applied so that for any link l in L, there is a k in L such that for each tuple $t_w = \ll w_i, l, w_j >, v_t >$ in the weighted tuple structure T_W, the tuple $t_w^{-1} = \ll w_j, k, w_i >, v_t >$ is also in T_W (k is the inverse link of l).

In a DM the weighted tuple structure is formalized as a labeled third-order tensor. Tensors are multi-way arrays, conventionally denoted by boldface Euler script letter: $\boldsymbol{X}$. The order of a tensor is the number of indices needed to identify its elements. An array with three indices is third-order tensor and the element (i, j, k) of a third-order tensor $\boldsymbol{X}$ is v_{ijk}.

A way to display third-order tensors is by nested tables where the three indices are respectively in the header column and in the two header rows. A fiber is equivalent to rows and columns in a high-order tensors and it is obtained by fixing the values of all indices but one.

A labeled tensor $\boldsymbol{X}_\lambda$ Is a tensor such that for each of its indices there is a one-to-one mapping on the integers from 1 to I (dimensionality of the index) to I distinct string (labels of the index). A weighted tuple structure T_W built from $W_1,\ L$ and W_2 can be represented by a labeled third-order tensor $\boldsymbol{X}_\lambda$with its three indices labeled by $W_1,\ L$ and W_2, respectively, and such that for each weighted tuple $t \in \ T_W = \ll w_1, l, w_2 >, v_t >$ there is a tensor entry $(i{:}\, w_1, j{:}\, l, k{:}\, w_2) = \ v_t$.

	j=1:own	*j=2:use*	*j=1:own*	*j=2:use*	*j=1:own*	*j=2:use*
	k=1:bomb		*k=2:gun*		*k=3:book*	
i=1:marine	40.0	82.1	85.3	44.8	3.2	3.3
i=2:sergeant	16.7	69.5	73.4	51.9	8.0	10.1
i=3:teacher	5.2	7.0	9.3	4.7	48.4	53.6

Table 1: Example of labeled third-order tensor (3 x 2 x 3)

Matricization is the operation that rearranges a high-order tensor into a matrix. The simplest case in mode-n matricization, which rearranges the mode-n fibers to be the columns of the resulting $D_n \times D_j$ matrix. In other words, in the case of a three-order tensor it makes vertical, horizontal or depth-wise slices of a three-way object and arranges these slices sequentially to obtain a matrix.

In DMs, the matricization is applied to labeled tensors: in the resulting labeled matrices row and column vector spaces correspond to the linguistic object that are studied. Such vectors can at this point be used to perform all standard linear algebra operation applied in vector-space semantics, such as measuring cosine similarity or applying matrix transformations for dimensionality reduction.

From the weighted tuple structure T_W of a DM, by matricizing the corresponding labeled third-order tensor $\boldsymbol{X}_\lambda$ four distinct semantic vector spaces can be obtained.

1. Word by link-word ($W_1 \times LW_2$)

	1:<*own*, *bomb*>	2:<*use*, *bomb*>	3:<*own*, *gun*>	4:<*use*, *gun*>	5:<*own*, *book*>	6:<*use*, *book*>
1:*marine*	40.0	82.1	85.3	44.8	3.2	3.3
2:*seargeant*	16.7	69.5	73.4	51.9	8.0	10.1
3:*teacher*	5.2	7.0	9.3	4.7	48.4	53.6

Table 2: Example of $W_1 \times LW_2$matrix of the tensor represented in Table 1

2. Word-word by link ($W_1 W_2 \times L$)

	1:*own*	2:*use*
1:<*marine,bomb*>	40.0	82.1
2:<*marine,gun*>	16.7	69.5
3:<*marine,book*>	5.2	7.0
4:<*sergeant,bomb*>	85.3	44.8
5:<*sergeant,gun*>	73.4	51.9
6:<*sergeant,book*>	9.3	4.7
7:<*teacher,bomb*>	3.2	3.3
8:<*teacher,gun*>	8.0	10.1
9:<*teacher,book*>	48.4	53.6

Table 3: Example of W_1 $W_2 \times L$ matrix of the tensor represented in Table 1

3. Word-link by word ($W_1L \times W_2$)

	1:*bomb*	2:*gun*	3:*book*
1:<*marine,own*>	40.0	85.3	3.2
2:<*marine,use*>	82.1	44.8	3.3
3:<*sergeant,own* >	16.7	73.4	8.0
4:<*sergeant,use* >	69.5	51.9	10.1
5:<*teacher,own* >	5.2	9.3	48.4
6:<*teacher,use*>	7.0	48.4	53.6

Table 4: Example of $W_1L \times W_2$ matrix of the tensor represented in Table 1

4. Link by word-word ($L \times W_1W_2$)

	1:<*mar., bomb*>	2:<*serg., bomb*>	3:<*teac., bomb*>	4:<*mar., gun*>	5:<*serg., gun*>	6:<*teac., gun*>	7:<*mar., book*>	8:<*ser., book*>	9:<*teac. , book*>
1:*own*	40.0	16.7	5.2	85.3	73.4	9.3	3.2	8.0	48.4
2:*use*	82.1	69.5	7.0	44.8	51.9	4.7	3.3	10.1	53.6

Table 5: Example of $L \times W_1W_2$ matrix of the tensor represented in Table 1

In space 1, attributional similarity can be calculated, for tasks like synonym detection of concept categorization, while in space 2 relational similarity among different word pairs can be measured. The other two matrices can be used for other semantic tasks like verb classification with space 3 or feature selection with space 4.

In conclusion, the DM framework allows the analysis of different kind of semantic spaces with the use of one single model, without requiring additional computational cost with respect to traditional DSMs.

1.3 Distributional memories for English

DMs approach has been experimented by Baroni and Lenci in 2010 with the creation of three different models for English[1].

Word-link-word tuples are extracted from a dependency-parsed corpus. The models are trained on the concatenation of the Web-derived ukWaC corpus[2], a mid-2009 dump of the English Wikipedia[3] and the British National Corpus[4]. The resulting concatenated corpus was tokenized, POS-tagged, and lemmatized with the TreeTagger5 and dependency-parsed with the MaltParser[5]. It contains about 2.83 billion tokens.

The label sets $W_1 = W_2$ contain 30,693 lemmas (20,410 nouns, 5,026 verbs, and 5,257 adjectives). These terms were selected by considering their frequency in the corpus. The words are stored in POS-suffixed lemma form.

The weighted tuple structures differ for the choice of links in L and/or for the scoring function σ. Therefore, the models differ in the degree of lexicalization of the links set and the type of weight chose.

1.3.1 DepDM

In this model, dependency paths are assumed to approximate well the semantic relations between words. The links set, L_{DepDM}, includes then only those, with the minimum degree of lexicalization among the three models, since the only lexicalized links are prepositions. Dependencies between words with more than five intervening items were discarded, in order to have a better reliability and filter out parsing errors.

The following *noun-verb*, *noun-noun* and *adjective-noun* links have been included:

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

[2] http://wacky.sslmit.unibo.it/doku.php?id=corpora

[3] http://en.wikipedia.org/wiki/Wikipedia:Darabase_download

[4] http://www.natcorp.ox.ac.uk

[5] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger

- **sbj_intr**: subject of a verb that has no direct object: *The teacher is singing* → <*teacher*, subj_intr, *sing*>;
- **sbj_tr**: subject of a verb that has a direct object: *The soldier is reading a book* → <*soldier*, sbj_tr, *read*>;
- **obj**: direct object: *The soldier is reading a book* → <*book*, obj, *read*>;
- **iobj**: indirect object in a double object construction: *The soldier gave the woman a book* → <*woman*, iobj, *read*>;
- **nmod**: noun modifier: *good teacher* → <*good*, nmod, *teacher*>M
- **coord**: noun coordination: *teachers and soldiers* → <*teacher*, coord, *soldier*>;
- **prd**: predicate noun: *The soldier became sergeant* → <*sergeant*, prd, *become*>;
- **verb**: an underspecified link between a subject noun and a complement noun of the same verb: *The soldier is reading a book* → <*soldier*, verb, *book*>;
- **preposition**: every preposition linking the noun head of a prepositional phrase to its noun or verb head: *I saw a soldier with the gun* → <*gun*, with, *soldier*>;

For each link, the inverse is also extracted and included. The cardinality of $L_{DepDM}$is 796.

The scoring function σ is provided by Local Mutual Information. Given the co-occurrence count O_{ijk} of three elements of interests (first word, link, second word) and the corresponding expected count under independence E_{ijk}:

$$LMI = O_{ijk} \log \frac{O_{ijk}}{E_{ijk}} \tag{1}$$

Avoiding overestimation of the significance low frequency events, negative weights are raised to 0. Non-zero tuples in the tensor is about 110M.

DepDM is a 30,693 x 796 x 30,693 tensor with density 0.0149% (proportion of non-zero entries).

1.3.2 LexDM

This model is heavily lexicalized and heterogeneous. Lexical material connecting two words is considered to be very informative about their relations. As a consequence, this model contains complex links, each with the structure "pattern + suffix".

The suffix is in turn formed by two substrings separated by a +, each encoding respectively of w_1 and w_2 features like POS, morphological features, presence of an article and of adjectives for nouns, presence of an adverbs for adjectives, presence of adverbs, modals and auxiliaries for verbs, diatheses of verbs. High frequency adjectives and adverbs are also contained in the string.

Link patterns, together with standard syntactic relations, include lexicalized dependency relations (specific verbs) and lexico-syntactic shallow templates. The latter are patterns used to extract specific pieces of semantic knowledge.

LexDM links set include DepDM's one, plus the following:

- **verb**: a list of 52 high frequency verbs can replace the verb link by the verb itself;
- **is**: copulative structures with an adjectival predicate (e.g. "*The soldier is tall*");
- **preposition-link_noun-preposition**: connecting expression such as "*a number of*", "*in a kind of*", with link_noun being one of 48 semi-manually selected nouns;
- **attribute_noun**: one of 127 nouns extracted from WordNet and

expressing attributes of concepts, such as “*size*”, “*color*”;

- **as_adj_as**: an adjective and a noun that match the template *as ADJ as (a|the) NOUN* (e.g. “*as sharp as a knife*”);
- **such_as**: two nouns occurring with the templates *NOUN such as NOUN* and *such NOUN as NOUN* (e.g. “*animals such as cats*”, “*such vehicles as cars*”).

The scoring function is the same as that in DepDM. The number of non-zero tuples is about 355M. LexDM is a 30,693 x 3,552,148 x 30,693 tensor with density of 0.00001%.

1.3.3 TypeDM

This model stems from the idea that the relevance of a link is not just a function of its frequency, but mostly of the variety of surface forms that express it (Baroni et al., 2010). For example, looking at the frequency of the triple <*fat, of*$^{-1}$*, land*> (a figurative expression) it appears to be much more common of the triple <*fat, of*$^{-1}$*, animal*>, which is instead more semantically informative. On the other hand, observing surface realizations in the corpus, the former has three while the latter has nine.

As a consequence, the links set consist in the patterns of LexDM links, while the suffixes of these patterns are used to count their number of distinct surface realizations. The scoring function σ computes LMI on the number of distinct suffix types displayed by a link when it co-occurs with the relevant words. Hence, the model do not counts tokens of realizations but types.

The number of non-zero tuples is about 355M. LexDM is a 30,693 x 25,336 x 30,693 tensor with density of 0.0005%.

2. Distributional memories for German

Syntax-based distributional semantics models are built more rarely than other kinds of representation of semantic information as they require accurate parsers and result in a high data sparseness. As a consequence, large-scale distributional models of this kind have been built for few other languages, beyond English.

However, different models of DMs for German have been built by Padò and Utt in 2012. According to their analysis, two strategies can be followed to build a new model for a language starting from the English DM:

1) **parallel induction**, that is the replication of the same schema used for the creation of the resource for English by Baroni and Lenci;
2) **crosslingual method**, which by translating the English DM benefits from the good quality of the resources used for that language, to overcome the absence or the lower quality of the resources available for other languages.

Through the first method, resources of quality comparable to the English DM have been obtained only for German and Croatian. As for the former, the construction was enabled by the presence of good parsers and large corpora. It was instead more difficult for Croatian due to resources scarcity.

Using these data, the effects of the second methodology, namely translation, have been evaluated, through experiments on the languages pairs.

The two models obtained for German have also been combined through a multilingual method of DM construction, thus joining data from the corpus of the target language and of the source language.

The three models have then been compared in terms of size, lexical coverage and performances in tasks of synonym detection and word similarity prediction.

2.1 Parallel induction: DM.De

DM.De[6] is the distributional memory for German built by Padò and Utt (2012) reproducing the schema of the English DM. DepDM variant was chosen because, differently from LexDM, it does not require manual annotations and gives almost always better results of the more complex TypeDM.

The types of links used correspond to the syntactic schemes of DepDM and are obtained via the observation of the most frequent syntactic configurations in a large German corpus. These can be divided into lexicalized and non-lexicalized patterns.

Non-lexicalized pattern:

- **sbj_tr**, **sbj_intr**: transitive and intransitive verbs subject;
- **obj**, **iobj**, **vcomp**: direct and indirect objects, phrasal complements of verbs;
- **nmod**: noun modification;
- **verb**: the relation between a subject and an object of a verb .

Lexicalized patterns:

- **n1 [prep] n2**: e.g. *Recht auf Auskunft* (en: “*right to information*”) → < *Recht*, Auf, *Auskunft* >;
- **adj n1 von [n2]**: e.g. *heutige Größe von der Sonne* (en: “*current size of the sun*”) → < *heutige*, *Sonne*, *Größe* >;

[6] `http://www.ims.uni-stuttgart.de/institut/mitarbeiter/uttjn/data.html`

- **n1 [verb] n2**: e.g. *Hochtief sieht Aufwind* (en: “*Hochtief* (i.e. a German construction company) *is succeeding* (idiom; literally “*sees the upwind*”)” → < *Hochtief*, *sehen*, *Aufwind*> .

German presents some difficulties when extracting word relations from text differently from English. Particle verbs for example often possess a detachable prefix e.g. *mit|geben*, which at the surface level can be realized at a large distance from the verb stem, e.g. *Er gab ihr das Buch, nach dem sie am Vortag gefragt hatte, nur ungern mit* (en: “*He gave her the book reluctantly, after she had asked the day before*”). Such verbs are reconstructed from the parser output. Addition issues are the very productive compounding (e.g. *Wasserstoffbetankungseinrichtung;* en: “*hydrogen-filling-station*”, literally “*Water-stuff-tanking-installation*”) and derivation (e.g. *Pappkärtchen;* en: “*paper card*”) of nouns. Much more noun types need thus to be integrated into the system than in English. As a consequence, differently from the English DM, no limits on the number of nouns, as well as the one of any other parts of speech, have been set, as this would make the tensor even sparser than usual in dependency-based representations.

Co-occurrences are extracted from the SDEWaC corpus[7], which is based on DEWaC, a wide collection of web texts belonging to the top-level domain .de. It consists in 9M word type and 884M word tokens. As for the syntactic analysis of the corpus the German dependency parser MATE[8] was used.

The weighting method for the German model is the same as the one used for the English one (LMI).

The resulting DM contains more than 78M link, 3.5M words (noun, verbs and adjectives) and 220K link types. This makes it much sparser than the English one (131M link, 31K words and 25K link types).

[7] http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/sdewac.en.html

[8] http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/matetools.en.html

2.2 Crosslingual method: DM.Xl

To overcome the problem of the absence of high quality resources for most of the languages, a method of translation of the DMs has been developed by Padò and Utt (2012). Using English as source language, it was obtained a model for the target language exploiting the wealth of processing techniques available for the former.

The crosslingual construction of DMs does not use a corpus of texts in the target language, neither monolingual or bilingual. It only used a translation lexicon: a list of lemma-translation pairs, without probabilities assigned. Resources like this one are extremely common and available even for languages that do not have large corpora. They are also often built through crowdsourcing and are available for download on the web.

The DM is analyzed as a directed graph and thus translation is expressed in terms of this structure. Ideally, if only one lemma in the target language corresponded to each lemma in the source language, so if the translation lexicon was a bijective function ($\mathrm{Tr}: S \rightarrow T$), the graph transformation related to the translation would just consist in relabeling the nodes with the expressions in the language that the model is made for. For each node in the model for English there would be a node in the one for the other language.

Since translation is instead a many-to-many relation, the complexity of the procedure increases. As a matter of fact, in a language like German there are an average of 2.3 translations for each English lemma and 1.9 translations from English for a German lemma.

The following functions determine translations respectively from the source language to the target language and viceversa:

(2)

$$\mathrm{Tr}: \mathrm{S} \rightarrow 2^{T}$$

$$Tr^{-1}: \mathrm{S} \rightarrow 2^{S}$$

A possible way to cope with this is using all the translations of a given word. The number of edges for each one in the source DM is $|Tr(s_1)| \cdot |Tr(s_2)|$.

(3)

$$E_T = \{(t_1, l, t_2) | \exists\, (s_1, l, s_2) \in E_S : \\ t_1 \in Tr(s_1) \wedge t_2 \in Tr(s_2)\}$$

The weight of the edge would be the average of the weights of all the edges in the source graph that refer to that.

(4)

$$\sigma_T(t_1, l, t_2) = \sum_{\substack{s_1 \in Tr^{-1}(t_1) \\ s_2 \in Tr^{-1}(t_2)}} \frac{\sigma_T(t_1, l, t_2)}{|Tr^{-1}(t_1)| \cdot |Tr^{-1}(t_2)|}$$

However, this method is problematic since the resulting graph contains an extremely high number of edges and because of a substantial loss in correctness of the DM. Moreover, lexical ambiguity also determines more than one possible translations. In a case like the English word *wood* two senses happen to exist for the translation in German: the sense of *forest*, translatable as *Wald*, and the one of *timber*, translatable as *Holz*. Considering two modification adjectives like *precut*, plausible for the sense of *timber*, and *great*, plausible for the sense of *forest*, if all the translations of the word *wood* are used without any filtering, both *Holz* and *Wald* would be linked to the two adjectives, thereby producing non-pertinent edges in the resulting DM.

A possible criterion for solving this is filtering by "backtranslation". Sticking to the previous example, *wood* will have two translations in German but the adjective *precut* would only have one. When backtranslating the two candidate edges obtain by the English edge <*precut* mod *wood*>, namely <*zeguschnitten* mod *Holz*> and <*zugeschnitten* mod *Wald*>, while the first one will only map to the original one, the second one will map to a different source edge, <*precut* mod *timber*>, being thus more probable than the other. To formalize this, a filtering condition is added to Equation 3: target edges must be among the highest-scoring edges for some source edge.

$$E_T \ = \{(t_1, l, t_2)|\ \exists\ (s_1, l, s_2) \in \ E_S : \\ t_1 \in Tr(s_1) \land t_2 \ \in Tr(s_2) \land \\ \sigma_T(t_1, l, t_2) = \max_{\substack{t \in Tr(s_1) \\ t' \in Tr(t_2)}} \sigma_T(t, l, t')\} \tag{5}$$

In monolingually constructed DMs, all links are by default used, as all the information is assumed to be reliable. Because of the unclear situation with the crosslingual DM, two ways of computing semantic similarity between vectors have been implemented. The first one, *AllL* uses the complete vectors; the second, *SPrf*L, uses only inverse links for verbs and regular links for nouns and adjectives. This stems from the assumption that selectional preferences are most informative and most likely to survive translation.

Observing the different results outcome by filtering the vectors or not on both the monolingual and crosslingual DM, as expected there is no difference between the two cases on the former, while higher precision is reached in the latter using the *SprfL* version. Consequentially, this condition is adopted in the crosslingual DMs.

2.3 Multilingual method: DM.MULTI

The crosslingual method of DM construction assumes that the resources in the target language are not good enough or are absent to build a DM in the traditional way. However, more corpora and parsers continually become available and combining monolingually and crosslingually constructed DMs would enable to merge corpus evidence both from the source and the target language. Padò and Utt (2012) therefore tried to combine the resulting semantic similarities produced by the two types of multilingual DMs, rather than the graphs themselves.

The possibility of a good quality linear interpolation of the models similarity is explained by the assumed complementary properties of the two models: while the monolingual model has a higher coverage, the crosslingual has higher quality.

Two strategies have been followed:

- **DM.MULTI Backoff**: this combination starts with the crosslingual model and falls back to the monolingual one in the case of zero-similarities;
- **DM.MULTI MaxSim**: the higher prediction between the one from the monolingual model and the one from the crosslingual is taken; therefore, both noise and sparse data are considered to underestimates similarities.

The two variants assume that the two models have the same score distribution. As there is no guarantee of that, the values are linearly transformed so that the resulting distribution has a mean of 0 and a standard deviation of 1.

2.4 Models evaluation

Experiments on the English-German DM pairs have been conducted in order to show the benefits of crosslingual and multilingual methods (Padò and Utt, 2012), following the setup of Mohammad et al. (2007).

The standard tasks chosen for the evaluation are synonym choice and prediction of human relatedness judgments. This way, two different aspects of lexical semantics are considered to test the models: a specific lexical relation and general semantic relatedness. The latter type of experiments and its results will be described here, as it is the same task that has also been used for the evaluation of dimensionality reduction methods.

As for similarities prediction, the Gur350 dataset[9] was chosen, that is a German relatedness dataset built by asking native speakers to assign a similarity judgment to a number of word pairs in order to test the performance of distributional similarity measures. It contains 350 word pairs tagged with a score of relatedness on a five-point scale between 0 (unrelated) and 4 (fully

[9] https://www.ukp.tu-darmstadt.de/data/semantic-relatedness/german-word-choice-problems/

related), as the mean of the scores assigned by the subjects. The results of the model obtained with the crosslingual and multilingual methods have been compared to previous works.

The procedure has implied the matricization of the DM into a word by link-word space ($W \times LW$) and the word similarities computation using Cosine similarity[10]. The correlation between the model predictions and the human relatedness judgments for word pairs is calculated by the Pearson's correlation coefficient[11].

Models have been compared in two condition:

1) *All* condition: the model makes a prediction on every item in the dataset;
2) *Covered* condition: cases of zero similarities are ignored.

Coverage have been calculated as the percentage of items with similarity greater than 0.

The models considered in the experiments are:

- Monolingual model: DM.De, constructed from SDeWAC (900M tokens), parsed with MATE, assuming *AllL* condition;
- Crosslingual model: DM.XL obtained by translation of the English TypeDM by Baroni and Lenci. Two versions are obtained with or without using backtranslation as a filter (respectively DM.XL *filter* and DM.XL *naive*).

[10] Cosine similarity is used as measure of word similarity, by looking at the distance between two word vectors.

$$\mathrm{d}(\overrightarrow{w_1}, \overrightarrow{w_2}) = \cos\theta = \frac{\overrightarrow{w_1} \bullet \overrightarrow{w_2}}{|\overrightarrow{w_1}|\,|\overrightarrow{w_2}|} = \frac{\sum_{i=1}^{n} w_{1_i} w_{2_i}}{\sqrt{\sum_{i=1}^{n} w_{1_i} w_{1_i}} \times \sqrt{\sum_{i=1}^{n} w_{2_i} w_{2_i}}}$$

[11] Pearson's coefficient is a measure of linear correlation between two variables. It is represented by the letter ρ and is calculated as:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

where cov is the covariance between two variables and σ is the standard deviation of a variable. As a result, $1 < \rho_{X,Y} < 1$.

The translation lexicon used is the community-built English-German `dict.cc`[12] online dictionary.

	Adjectives	Nouns	Verbs	Total
English	37K	78K	8K	123K
German	35K	99K	9K	143K
Translations pairs	77K	172K	28K	277K

Table 6: Size of the `dict.cc` dictionary

A drawback of this choice is that, despite of its large size and coverage, as Table 6 shows, the lexicon contains relatively few verbs, so many verbal data have been excluded. The *SprfL* condition is assumed.

- Multilingual model: DM.MULTI Backoff and DM.MULTI MaxSim, each combining DM.De and DM.Xl filter;
- Bag-of-words models: A standard BOW model has been built using the same German corpus used for DM.DE. A window of 10 context words to the left and right is assumed; the dimensions consist of the top 10K most frequent content words (nouns, adjectives, verbs and adverbs) . Another word-based model (BOW PCA_{500}) created reducing the other to 500 dimensions by applying principle component analysis.
- Models from the literature: the state of the art is represented by the monolingual ontology-based models that use GermaNet, (German) Wikipedia or both (Lin_{GN}[13], JC, PL[14]) and crosslingual distributional models that represent the meaning of German lemmas in terms of English thesaurus categories (Lin_{dist}[15]).

[12] `http://www1.dict.cc/translation_file_request.php?l=e`

[13] Mohammad et al. (2007)

[14] Zesch et al. (2007)

[15] Mohammad et al.(2007)

Class	Model	Nodes	Edges
Monolingual	DM.De (DE)	3.5M	78M
	TypeDM(EN)	31K	131M
Crosslingual	DM.Xl *naive*	63K	5B
	DM.Xl *filter*	63K	1.7B

Table 7: Sizes of the different DMs

As Table 7 shows, because of the larger English corpus and the inclusion of low-frequency items in DM.DE, the English DM is much more compact and denser than the German one. The crosslingual models have twice the English coverage but two orders of magnitude below the monolingual DM.DE. Filtered translation has a consistent effect on the reduction of the size of the DM, considering that in the *filter* version the number of edges is increased with translation by a factor of 13 while *naive* imply a factor of 30. The problem of the overgeneration is only partially solved.

Model	*All*	*Covered*	
	Correlation	Correlation	Coverage
Baselines and word-based DSMs			
Frequency	.13	.13	1
BOW	.20	.21	.97
BOW PCA$_{500}$	.34	.37	.97
Syntax-based DSMs			
DM.De	.38	.43	.60
DM.Xl EN → DE naive	.29	.38	.61
DM.Xl EN → DE filter	.33	.49	.49
DM.MULTI Backoff	.40	.45	.69
DM.MULTI MaxSim	.42	.47	.69
Models from literature			
Lin$_{GN}$	NA	.50	.26
Lin$_{dist}$	NA	.51	.26
JC$_{GN}$ + PL$_{WP}$	NA	.59	.33

Table 8: Correlation and coverage values in word similarity prediction on the Gur350 dataset

Results in the task from all the different models considered can be observed in Table 8 above and summarized as follows:

- DM.De outperforms the BOW model though not consistently and with a decrease in coverage.
- DM.Xl has instead the highest value of accuracy among all the syntax-based models. Backtranslation filter brings a strong improvement and on the other hand causes a lower coverage.
- DM.MULTI almost reaches the quality of DM.Xl and has the highest coverage among the models of its class. Between the two versions, MaxSim performs better.
- DMs models have less accuracy than models from literature, but a higher coverage.

As a consequence, building crosslingual DMs, not relying on the use of target language corpora, seems represent a valid alternative to the “parallel induction” methodology, as the performances of these models are closer or even better than the monolingual DMs ones. In case that monolingual model of this kind are available, combining this data in order to get a multilingual one is a consistent advantage over both monolingual and crosslingual model.

In comparison with models from literature, though DMs has a higher coverage of hand-constructed knowledge as we can expect, they performs worst. However, by imposing a threshold towards infrequent events it has been proved possible to reach an accuracy which is equal to ontology-based models (.59). Efficiency issues and the possibility of improving performances has directed research towards experiments of dimensionality reduction on syntax-based distributional semantics models.

3. Dimensionality reduction

When dealing with semantic similarity calculations, dimensionality reduction, or factorization, is an operation that enables to find a smaller number of uncorrelated or lowly correlated dimensions in semantic models (Van de Cruyus, 2010). The reasons to apply this transformation to the data are:

- reducing a large feature space to a much smaller number of dimensions, in order to strongly decrease the computational cost of similarity calculations;
- discovering latent structure in the data, as factorization is able to generalize over individual data samples and overcome data sparseness and noise.

More than one methods to apply this transformation to models are possible. Singular Value Decomposition (SVD) is the underlying operation of one of the most famous dimensionality reduction methods, namely Latent Semantic Analysis (LSA). Non-negative Matrix Factorization (NMF) is another dimensionality reduction algorithm that overcome some issues correlated with LSA.

3.1 Singular value decomposition

Singular Value Decomposition is generally correlated to Latent Semantic Analysis (Landauer and Dumais, 1997), which models the meaning of words and contexts in general (especially documents) by projecting them into a vector space of reduced dimensionality. This reduction is applied by the linear algebraic method of singular value decomposition to a simple term-by context frequency matrix. By enforcing a lower number of dimensions, the algorithm is forced to make generalizations over the data. Co-occurring terms are mapped to the same dimensions; terms that do not co-occur are mapped to different dimensions.

SVD is often used in statistical applications in different scientific fields, such as image recognition, signal processing and information retrieval. This operation can also be used for a DM applying SVD to the word-by link-word matrix obtained by the tensor.

In linear algebra, a rectangular matrix can be decomposed into three other matrices such that their product is equal to the original matrix:

$$A_{m\times n} = U_{z\times z}\,\Sigma_{z\times z}(V_{n\times z})^T \tag{6}$$

where $z = min(m, n)$.

A graphical representation of SVD is given below in Figure 1.

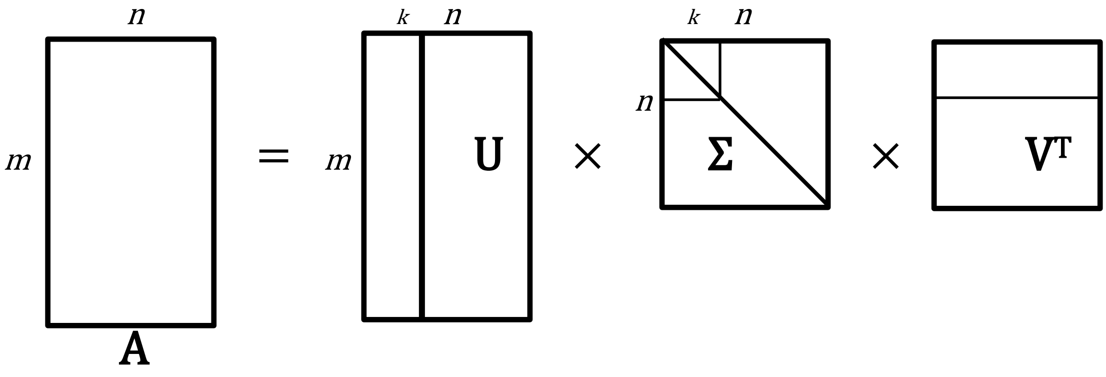

Figure 1: Graphical representation of SVD

- A is the original matrix of size $m \times n$;
- U is an $m \times z$ matrix that contains newly derived vectors, called left-singular vectors;
- V is an $n \times z$ matrix of derived vectors, called right-singular vectors;

- Matrix V^T consists in the transpose of matrix V;
- Σ is a $z \times z$ square diagonal matrix (that is with non-zero entries only in the diagonal), that contains derived constants, called singular values;
- For all the derived vectors, all the dimensions are orthogonal (i.e. linearly independent) to each other, so that each dimension is uncorrelated to the others.

SVD can be seen as a method of rotating the axes of the n-dimensional space so that the largest variation is captured by the leading dimensions. The diagonal matrix Σ contains the singular values sorted in descending order; each value represents the amount of variance that is captured by a particular dimension. The left-singular and right-singular vector linked to the highest singular value is the most important dimension in the data; the singular vectors linked to the second highest value is the second most important, and so on. Typically, only $k \ll z$ dimensions are used; in this way the least significant singular values are omitted.

Thus, SVD is able to transform the original matrix, with its overlapping dimensions, in a new smaller one that describes the data as its principle components, resulting in a more succinct and general representations. As a consequence, it provides a filter for redundancy and a reduction of data sparseness.

A drawback of SVD is the fact that its probabilistic interpretation assumes data to be normally distributed. This is not the case of frequency count data, thus the reconstruction of a matrix may contain negative numbers. It is not clear what these kind of values on a semantic scale should designate.

3.2 Non-negative matrix factorization

Differently from SVD, Non-negative Matrix Factorization (Lee and Seung, 2000) stems from the key idea of imposing a non-negativity constraint on the factorization. This implies a parts-based representation (that is a part is reconstructed as linear combination of the different parts), as only additive combinations are allowed. This often results in more distinct and clear characteristics extracted from the data.

NMF is popular in fields such as image recognition, speech recognition and machine learning. Though less common in distributional semantics, this type of factorization can be also seen as a possible dimensionality reduction method of DMs, if applied to the word-by link-word matrix obtained by the tensor.

A group of algorithms is identified by the name NMF: in all of these a matrix V is factorized into two other matrices, W and H.

$$V_{n\times m} \approx W_{n\times r} \times H_{r\times m} \tag{7}$$

A representation of NMF is given in Figure 2.

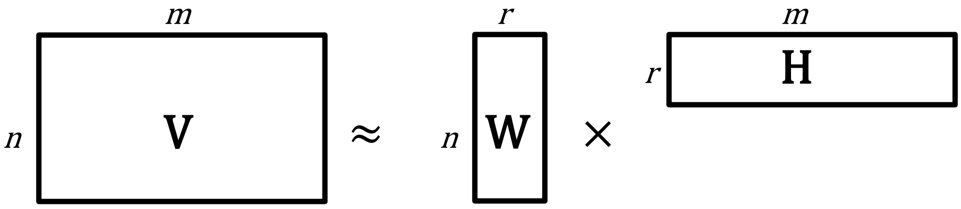

Figure 2: Graphical representation of NMF

Typically, r is much smaller than n, m and both instances and features are expressed with few components. The non-negative constraint enforces that all elements must be greater or equal to zero.

Two objective functions can be used to quantify the quality of the approximation of the original matrix: one minimizes the sum of squares, that is a measure of the variance, and one the Kullback- Leibler divergence, that is a measure of the difference between two probability distributions. In practice, matrix W and H are randomly initialized and update rules are iteratively applied, alternating between them. In each iteration the two matrices are normalized. The algorithm stops after a fixed number of iterations, or according to some stopping criterion. The update rules are guaranteed to converge to a local optimum, thus repeatedly running NMF algorithm result in the global optimum.

4. Dimensionality reduction on a crosslingual distributional memory

DMs obtained via translation raise issues related to their size: if we look at these models in terms of graphs, as Table 7 shows, they present at least one order of magnitude more edges than monolingual ones and significantly less nodes. Therefore crosslingual DMs result in extremely dense graphs. As a matter of fact, the ratio between words and entries in the model is much higher than in the English model.

On the one hand, this is due to the fact that, though these models are based on an English corpus and thus partially automatically compiled, they also rely on a manually compiled translation dictionary, which does not provide a high level of lexical coverage. This problem explains the low number of nodes (terms). On the other hand, backtranslation is only a partial solution to the lexical ambiguity that arises when translating from a language to another: even if backtranslation provides benefits to the model, still the number of edges increases when turning the model from English to German.

Dimensionality reduction methods have been applied during the project in order to smooth the crosslingual DMs and improve their efficiency and performances.

However, matrix factorization is not the only type of reduction that could have been chosen. Other methods like tensor and graph sparsification and bloom filtering have been considered as alternatives. While the formers would have affect the size by transformations on the model itself, the other would have stored the DMs in a data structure less expensive in terms of memory. Anyway, matrix transformations have been implemented and evaluated as a first attempt since they are the most common methods of size reduction in distributional semantics models.

In order to investigate the general effects of dimensionality reduction on distributional semantics models, at first this reduction method has not been tested directly on DMs but on simpler co-occurrences matrices derived from text corpora.

4.1 Methodology and tools

The application of dimensionality reduction has consisted in the following steps:

1) Building the semantic space, based on co-occurrence counts extracted from text corpora or values from the distributional memory after it has been metricized in the $W \times LW$ version;
2) Applying matrix transformations, namely SVD and NMF, on the space, reducing the matrix to 500 dimensions;
3) Testing the quality of the effect of the reduction in a task of word similarity prediction:
    - given a list of similarities judgments, calculating the similarity of the subset of words contained in the model, measured with Cosine Similarity, in order to compare the values;
    - observing the correlation between human judgments and the cosines computed by each model by plotting the values and looking at Pearson's correlation coefficient.

Dimensionality reduction has been implemented with Python programming language and as support a library specifically oriented to distributional semantics has been used, namely Dissect (Distributional Semantics Composition Toolkit)[16]. This library is part of the European Research Council project (2011- 2015) COMPOSES (Compositional

[16] http://clic.cimec.unitn.it/composes/toolkit/index.html

Operations in Semantic Space) and it is oriented to build and explore computational models based on the principle of distributional semantics, with particular focus on compositional meaning (Dinu, Pham, Baroni, 2013). It can be used for building semantic spaces out of co-occurrence matrices, applying transformations, weighting schemes and compositional operations on these and measuring semantic similarities. The library is based on Python’s Numpy and Scipy modules and thus optimized for speed.

With the Dissect library a semantic space can be built from a matrix in the `sm` (sparse matrix) or `dm` (dense matrix) format representing co-occurrences. This means that the input must consists in three files: a row file, a column file and a file containing co-occurrence counts or any other values associated with the cells. They must have the same name and different file extensions (`.rows`, `.cols`, `.sm`/`.dm`).

The row file consists of a list of strings, each corresponding to a row in the matrix. In the same way the column file represents the columns of the matrix. The matrix file in the dense format contains row strings followed by their associated vector. In the sparse format instead each line consists of three values: the row string, the column string and the count, so that only non-zero values are represented.

For example, given the co-occurrence matrix:

	toy	*tv*	*book*
man	3	5	0
woman	0	5	6
child	43	0	0

the files required by Dissect in order to create its semantic space would be:

- Row file (`.rows`):

```
man
woman
child
```

- Column file(`.cols`):

```
toy
tv
book
```

- Matrix file – Dense format (.`dm`):

```
man 3 5 0
woman 0 5 6
child 43 0 0
```

- Matrix file – Sparse format (.`sm`):

```
man toy 3
man tv 5
woman tv 5
woman book 6
child toy 43
```

For its size and consequent data sparseness, sparse matrix format was chosen for the implementation of the matrix factorizations on DM and for the preliminary testing.

Dimensionality reduction is then applied using again Dissect, that provides both the operations of SVD and NMF.

The functionalities included in the library that have been mainly used are:

- `composes.semantic_space.space.build`: this method reads in data files and extracts the data to construct a semantic space; it takes as input the three files of the required `sm` or `dm` formats and the specification of the format chosen itself;

- `composes.transformation.dim_reduction.svd.Svd`: Singular Value Decomposition to a reduced dimension k, specified as argument, is performed. Given an input matrix X, it computes the decomposition:

$$A = U \times \Sigma \times V^T$$

  and returns $U \times \Sigma$ truncated to dimension $\min\left(k, rank(A)\right)$;

- `composes.transformation.dim_reduction.nmf.Nmf`: this method performs Non-negative Matrix Factorization to reduced dimension *k,* specified as argument. Given as input a non-negative matrix X, it computes the decomposition $A \approx W \times H$ and returns the matrix W;
- `composes.semantic_space.space.get_sim`: this method computes the similarity between two words; it takes as arguments the terms themselves and the selected measure of similarity (in this case `CosSimilarity`).

Semantic spaces, including the ones resulting from the transformations, can be saved in *pickle* format, so that they can be reloaded in Python again for a later use. Word similarities calculation can be made by comparing the row vectors of the words: in case of no dimensionality reduction they can be extracted by looking at the simple matrix; in case of SVD U matrix is considered, and W for NMF.

In order to investigate the quality of of the semantic models, a set of similarity judgments has been chosen and then word similarity has been calculated only for pair of words belonging to that set. The resulting calculations have then been saved for each space in a file in `sims` format where each line consists in the pair of words and the cosine similarity value[17].

The cosines produced by the models can then be compared with each other and with human judgments. For this task, the R statistical environment has been used, reading the similarities files as tables. Both *All* and *Covered* condition, like with the evaluation of DM.Xl, have been considered: thus, measures have been computed both considering all the values and only considering those greater than 0. The correlation indexes between the values assigned by the standard space and its reduced versions and between those assigned by each space and the dataset, are calculated as Pearson's coefficient[18]. In the *Covered* case, the coverage percentage is also given.

Before Dissect, another Python library oriented to scalable statistical

[17] See Note 10

[18] See Note 11

semantics, namely Gensim[19], has been at first experimented with. Given text as input, Gensim build an object Corpus and then its vector space. The resulting matrix can be turned into a *numpy* or *scipy* format matrix. By doing that Numpy for SVD and another library like for example Pymf[20] could then be used to apply the transformations. Gensim functionalities have then been tested on the Brown Corpus. However, using Dissect has proved to be a better approach, since it provides all the tools needed for the experiments in a single library.

4.2 Preliminary experiments

4.2.1 Testing on English window-based models

In order to investigate the effects of different methods of factorization on co-occurrence matrices, a portion of the Brown Corpus[21] (30K tokens) has been used as input data and at a later stage the entire text collection (around 1M tokens). The Dissect toolkit has been used for building the semantic space reading from a sparse matrix representing co-occurrences in the corpus within a context window of 3 words. Later matrix transformation has been applied.

The word pairs whose similarity has then been calculated were taken from the WordSimilarity-353 Test Collection[22]. It contains two sets of English word pairs along with human-assigned similarity judgments. The first set contains 153 word pairs along with their similarity scores assigned by 13 subjects. The second contains 200 word pairs, with their similarity assessed by 16 subjects. Subjects had a near-native command of English and their instructions were to estimate the relatedness of the words in pairs on a scale from 0 (totally unrelated words) to 10 (very much related or identical words).

[19] https://radimrehurek.com/gensim/

[20] https://code.google.com/p/pymf/

[21]http://www.essex.ac.uk/linguistics/external/clmt/w3c/corpus_ling/content/corpora/list/private/brown/brown.html

[22] http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/

For our experiments, we used the the full list of 353 words, along with their mean similarity scores,.

Pearson's correlation indexes between the standard model and the reduced ones considering the *Covered* condition is 0.98 for SVD and 0.76 for NMF, with the former assigning very close values to the original. As expected, the value between each of the models and similarity judgments is instead not high due to the small size of the data, which makes predictions not sufficiently reliable.

The computation of SVD is much quicker than NMF which takes at least two hours, with a part of the algorithm execution independent from the data size and then of constant time (in the case of the entire Brown corpus, for example, working with equal infrastructure, there is between SVD and NMF a ratio of about 1':40').

The same process has then been applied to a larger collection of texts in order to observe the quality of the data after the reduction on a more consistent model. For this aim, the same corpus used for building the English DMs by Baroni and Lenci was used as input data, that is the concatenation of the British National Corpus (about 95M tokens), ukWAC (about 1.9B tokens) and English Wikipedia corpus (820M tokens). A context window of 5 words has been considered for each term in order to extract co-occurrences.

In this case, two datasets of judgments have been used: WordSim353 and MEN[23]. The latter is composed by two sets of English word pairs (one for training and one for testing) together with human-assigned similarity judgments, obtained by crowdsourcing (only native speakers). It consists of 3000 word pairs, randomly selected from words that occur at least 700 times in a large corpus, and sampled so that they represent a balanced range of relatedness levels according to a text-based semantic score. Each pair was more or less related than the comparison point by a subject, randomly matched with a comparison pair. Rather than ask annotators to give an absolute score reflecting how much a word pair is semantically related (like in Wordsim353), binary comparative judgments of relatedness are asked about two pair exemplars at a time to make the task much simpler for the annotator.

[23] http://clic.cimec.unitn.it/~elia.bruni/MEN.html

Each pair was rated against 50 comparison pairs, thus obtaining a final score on a 50-point scale. Since each pair presents values in a random order, all the 3000 have been associated with a standard 1-7 Likert scale.

Two types of semantic spaces have been built, one considering also the part of speech. In this case the correlation with similarities dataset is done with a POS-tagged version of MEN.

Corpus: BNC+ ukWac + Wackypedia $_{\text{no POS}}$ Word similarity prediction (Wordsim353)					
Semantic space	*All*		*Covered*		
	Correlation		Correlation		Coverage
	Standard space	Dataset	Standard space	Dataset	
Standard	-	.05	-	.30	.91
SVD	.99	.05	.98	.31	.91
NMF	.88	.16	.78	.30	.91

Table 9: Correlation values (Covered condition) of BNC+ Wac + Wackypedia $_{\text{no POS}}$ semantic space and its transformed versions with the original space and Wordsim353 dataset

Corpus: BNC+ ukWac + Wackypedia $_{\text{no POS}}$ Word similarity prediction (MEN)					
Semantic space	*All*		*Covered*		
	Correlation		Correlation		Coverage
	Standard space	Dataset	Standard space	Dataset	
Standard	-	.09	-	.27	.77
SVD	.99	.09	.97	.27	.77
NMF	.96	.14	.81	.27	.77

Table 10: Correlation values (Covered condition) of BNC+ Wac + Wackypedia $_{\text{no POS}}$ semantic space and its transformed versions with the original space and MEN dataset

Corpus: BNC+ ukWac + Wackypedia $_{POS}$ Word similarity prediction (MEN)					
Semantic space	*All*		*Covered*		
	Correlation		Correlation		Coverage
	Standard space	Dataset	Standard space	Dataset	
Standard	-	.13	-	.29	.92
SVD	.99	.12	.98	.29	.92
NMF	.84	.24	.79	.31	.92

Table 11: Correlation values (Covered condition) of BNC+ Wac + Wackypedia $_{POS}$ semantic space and its transformed versions with the original space and MEN dataset

As Table 9, 10 and 11 show, considering both the Wordsim353 and the MEN datasets, the correlation between similarities values of the standard semantic space and its transformed reduced version is consistent, especially in the case of SVD. NMF instead always tends to assign lower values than the standard space and SVD when looking at the similarities files.

Word pair	MEN $0 \leq sim \leq 50$	Standard $0 \leq sim \leq 1$	SVD $0 \leq sim \leq 1$	NMF $0 \leq sim \leq 1$
automobile-n car-n	50	0.95	0.96	0.93
eye-n smile-n	25	0.81	0.82	0.52
bakery-n zebra-n	0.0	0.93	0.95	0.88

Table 12: Examples (from MEN dataset) of word similarity values assigned by the spaces

The correlation with human judgments is not particularly relevant for the evaluation at this point since it is related to a different kind of distributional semantics model (some examples of similarity values can be observed though in Table 12), while the fact that it remains constant or improves with the dimensionality reduction of the model is instead interesting. However, in *All* condition all of the models perform drastically worse, while they get a strong improvement when considering only covered elements.

Since the coverage of the models is high, this may due to the dense presence of many highly related pairs of words in the datasets, that cannot have a high level of correlation with the values predicted by the model when it abstains for the absence of information about those and assign a value of 0. Anyway, these results are satisfactory if compared with the state of art of word-based distributional semantics models since the models' performances in word similarity prediction reach similar levels[24].

In conclusions, with these preliminary experiments, the methodology and the toolkit chosen for the implementation of matrix reduction have been tested, producing positive results. It has also been proved that in word similarity prediction, at least in simple distributional models, SVD and NMF gives good results, especially the former, as they do not affect performances negatively.

4.2.2 Testing on distributional memories: English DM

Having observed the general effects of these dimensionality reduction methods on simple distributional semantics models, the method has then been tested on the more complex target data structure. The procedure chosen for the task has been:

1) Selection of a number of most relevant link-word pairs in the DM model;
2) Reduction of the model filtering the entries of the DM given this subset of pairs;
3) Matricization of the reduced distributional memory in the sparse matrix $W \times LW$;
4) Building of the standard (that is not transformed) semantic space and its SVD and NMF versions.

The first step consists in a first rough size reduction of the model. Link-word pairs represent the contexts in the $W \times LW$ matrix, thus adding the

[24] Baseline and word-based DSMs in Table 8

condition that only the main ones in the model are kept and the less consistent are filtered out, already reduces the dimensions of the semantic space in a simple way.

More than one criterion to select the top link-word pairs are possible using:

- the frequency of the pair link-word, calculated by simple occurrence counts within the DM;
- the sum of the values associated with the pair, calculated by summing all the values of the tuples where it appears.

The first method entails relying on the fact that a semantic space that wants to model words' meaning on the basis of their distribution over a set of contexts may be considered more credible if it uses as dimensions for word vectors the most recurring contexts, which are the ones used with the largest range of terms; then to judge the importance of a link-word pair the number of words it co-occurs with is more important than the weight related to those association. On the other hand, the second method selects the contexts that appeas with a high number of words and with high values of association (LMI).

The differences between the two context subsets obtained using this two criteria of relevancy have been observed, using as test data the English TypeDM and as dissimilarity measure the Jaccard coefficient[25]. The value between the two subsets has been calculated considering an increasing number of n elements ($n \leq 5000$).

Number of links	5	10	50	100	200	500	1000	2000	5000
Jaccard's coefficient	.66	.81	.88	.92	.92	.97	.97	.97	.97

Table 13: Jaccard's coefficient between the subsets obtained via the two criteria of relevancy considering an increasing number of n elements.

[25] The Jaccard coefficient is an index of similarity between two sets. It is calculated as:

$$J(A,B) = \frac{A \cap B}{A \cup B}$$

Consequently, $0 < J(A,B) < 1$.

The more the cardinality of the subsets is increased the more the value of similarity between them gets close to 1. Therefore, choosing one or another does not make any substantial difference since the two cases seem to coincide very often. Apparently, when calculating the sum of the values, the number of sums (frequency) is much more influential than the weight itself and the most frequent pairs will tend to be assessed as the most relevant ones even with this criterion. Thus, the more simple frequency-based criterion has been chosen for filtering.

Because of the big size of the German DM, in a previous step the reduction has been implemented and tested on the English TypeDM.

The top 50K link-word pairs have been extracted from the data and a new reduced DM has been created including only the entries with these elements, decreasing the number from 131M to 46M.

The tensor has been matricized in the $W \times LW$ version and SVD and NMF have then been applied to the semantic space. Then, from 60K dimensions it has been reduced in both of the transformations to 500.

Calculating the similarities for the MEN dataset (with POS) with these models and comparing them to human judgements, results have been evaluated.

Reduced (top 50K link-word pairs) TypeDM Word similarity prediction (MEN)					
Semantic space	*All*		*Covered*		
	Correlation		Correlation		Coverage
	Standard space	Dataset	Standard space	Dataset	
Standard	-	.49	-	.50	.86
SVD	.95	.48	.94	.49	.86
NMF	.40	.14	.62	.17	.86

Table 14: Correlation and coverage values in word similarities prediction of semantic spaces derived from TypeDM (top 50K link-word pairs version) and its reduced versions

As Table 14 shows, the results obtained by these experiments suggests that:

- NMF performs badly in comparison with SVD in both *All* and *Covered* condition and its values of correlation with human judgements do not reach sufficiently good values.
- SVD semantic space results are very similar to those of the standard one, with a good value of correlation between each other and an almost equal performance in the task of word similarity prediction.
- The level of coverage is consistently good for a syntax-based model and it remains constant with the reduction.[26]

Word pair	MEN $0 \le sim \le 50$	Standard $0 \le sim \le 1$	SVD $0 \le sim \le 1$	NMF $0 \le sim \le 1$
automobile-n car-n	50	0.54	0.74	0.98
eye-n smile-n	25	0.22	0.30	0.99
bakery-n zebra-n	0.0	0.04	0.11	0.99
city-n town-n	39	0.91	0.95	0.99

Table 15: Examples (from MEN dataset) of word similarity values assigned by the spaces derived by the reduced TypeDM

Looking at similarity measurements, it is possible to observe that NMF space tends to assign in this case much higher values (very close to the maximum value of 1) than the ones assigned by the other spaces. As Table 15 shows, this happens for items with strong similarity but also with medium and low (even zero). Though this also happens for really similar pairs too, this unpredictable behavior of NMF explains the low correlation with the standard version of the model.

[26] The decrease in coverage in comparison with the original model is impossible with SVD and NMF, since the transformations only affect dimensions and cannot reduce the number of rows (words) in the matrix. The value of coverage (number of non-zero similarities) may grow instead since, reduced models will tend to assign zero-similarities only to uncovered elements and never to elements that do not actually share any context.

Thus, from this first testing of dimensionality reduction on a DM structure, SVD seems to be highly preferable as method of size reduction to NMF, whose results seem to be unpredictable and in this case its space does not even assign values that are close to those of native speakers. SVD similarity values instead seems to reduce data without a substantial loss in quality in comparison with the original model.

4.3 Matrix factorization applied to DM.Xl

Eventually, the dimensionality reduction has been applied to the target data: the crosslingual model DM.Xl *filter*. Due to the features of the German distributional memory, some further conditions have been added to the method experimented with English.

First of all, the structure of the entries in the model is slightly different from the English DM because of the translation itself. As a matter of fact, since the score is assigned to the triple as the mean of the scores of the entries in the source language that map to it according to Equation 4, each entry consists of the word-link-word tuple, the sum of the scores and the number of those. Then, when building the sparse matrix out of the reduced DM, the cell consists of the sum of scores divided by the number of scores (mean score).

In addition, because of the *SprfL* condition, in the crosslingual DM only inverse links for verbs and only regular links for nouns and adjectives are included. As a consequence, when building the reduced DM the entries where verbs occur as first word have been ignored and wherever a verb occurs as the second word the link has been reversed (first word inverted with the second) and flagged with '-1'.

Due to the content of the translation lexicon used for the translation, the crosslingual DM does not include much verbal data[27]. Thus, reducing the model with the filter of the top link-word pairs would result in a reduced DM containing very few verbs in comparison with a very high number of nouns.

[27] See Table 1

To cope with this drawback and have a better balance among the parts of speech in the reduction, instead of the top 50K link-word pairs, the top 20K respectively for nouns, adjectives and verbs have been selected and then joined as the top 60K of the DM.

Beyond these conditions, the same methodology tested for the English DM have been used for the target German DM.

4.4 Results evaluation

The original DM.Xl contains 1.7B entries, while the version reduced to the top 60K link-word pairs consists in 107M. Both SVD and NMF reduce the space derived from this model to 500 dimensions.

The dataset of human similarity judgements (Gur350) and the parameters (correlation and coverage) used for the evaluation of the models have been the same as those used for the experiments in word similarity prediction tasks with DM.Xl[28], in order to be able to judge the quality of the size reduction in comparison with the original.

Word similarity prediction (Gur350)					
Semantic space	*All*		*Covered*		
	Correlation		Correlation		Coverage
	Standard space	Dataset	Standard space	Dataset	
Reduced (top 60K link-word pairs) DM.Xl *filter*					
Standard	-	.29	-	.42	.50
SVD	.84	.24	.80	.31	.58
NMF	.72	.24	.60	.26	.57
Dm.Xl *filter*	-	.33	-	.49	.49

Table 16: Correlation and coverage values in word similarities prediction of semantic spaces derived from DM.Xl (top 60K link-word pairs version)and its reduced versions

[28] See Chapter 2.4

Table 16 summarizes the results given in the task of word similarity prediction. We can observe that:

- In *All* condition the correlation with the standard space is high, though not as for the English DM experiments, and better than *Covered* condition. That is explainable by the close numbers of uncovered elements among the spaces, whose zero-similarity values of are shared by all of them. As a consequence of this, the correlation with the dataset of human judgments is not particularly good.
- In both the conditions, the correlation with the standard space is better, as expected from the previous experiments, for SVD than NMF.
- In *Covered* case, SVD also gets a better result in terms of performance in the task, even though in comparison with the previous experiments the difference from the original model is bigger.
- As for the standard space, with the reduction to the top link-word pairs, the coverage is similar to the one of DM.XI *filter*. The correlation with Gur350 instead is 0.07 lower.
- With respect to the original space, there is a decrease of .09 with SVD space, which makes its quality inferior to the ones of the other DMs available for German[29]. NMF version scores even worse, though it reaches a higher coverage than the standard, like SVD.
- The results can be compared with those of other semantic models shown in Table 8. These spaces gets better coverage but worse performances than models from literature. The values of correlation are higher to those assigned by baseline and word-based ones (but with much less coverage) but lower than the other DMs available.

The distribution of the values of each transformed spaces with respect to the ones of the standard one can be observed as a diagram of the correlation between the two sets of values of similarity.

The plots showing the correlation between respectively SVD and NMF, and the original space, in the *Covered* condition, are reported.

[29] See Table 8

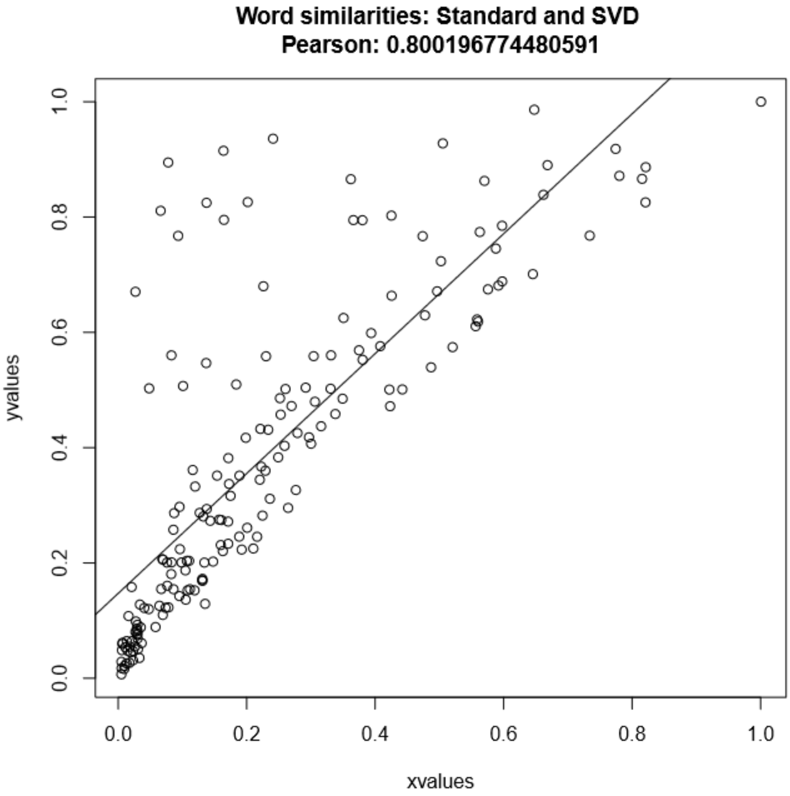

Figure 3: Diagram of the correlation between the standard version of the reduced DM.XI and its SVD version

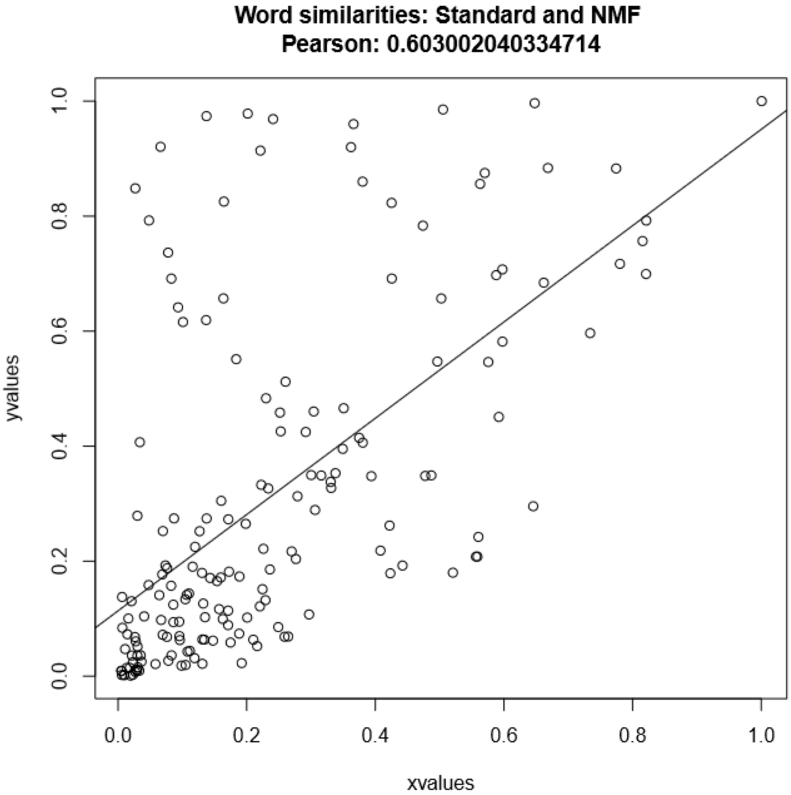

Figure 4: Diagram of the correlation between the standard version of the reduced

In the same way, the distribution of the values of each space with respect to the ones of the dataset can be observed as a diagram of the correlation between the cosines and the human similarities.

The plots of word similarity prediction of the three semantic spaces in the *Covered* condition are reported below:

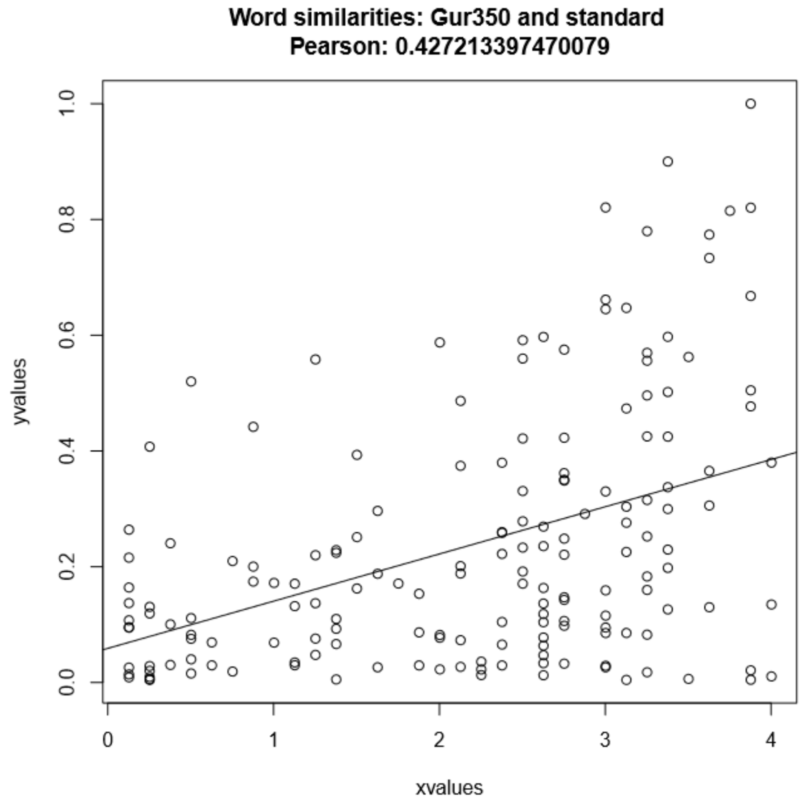

Figure 5: Diagram of the correlation between the standard version of the reduced DM.Xl and Gur350 dataset in word similarity prediction

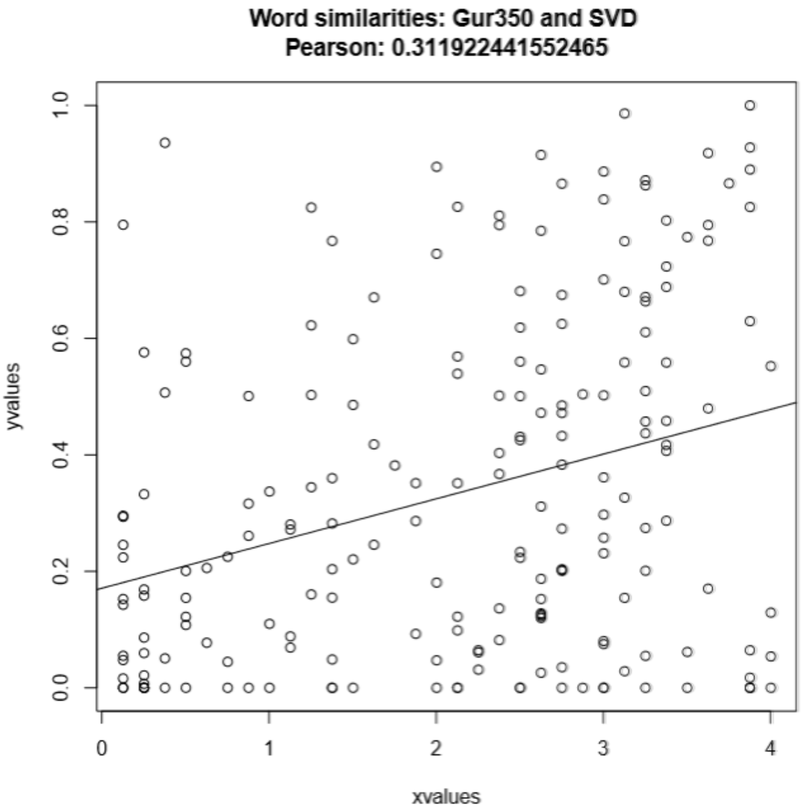

Figure 6: Diagram of the correlation between the SVD version of the reduced DM.Xl and Gur350 dataset in word similarity prediction

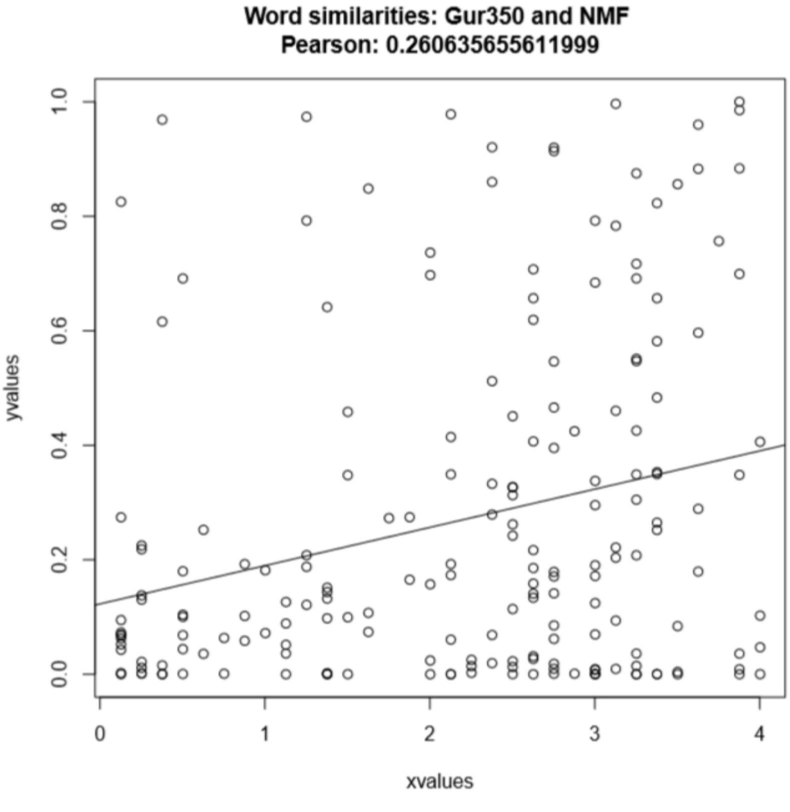

Figure 7: Diagram of the correlation between the NMF version of the reduced DM.Xl and Gur350 dataset in word similarity prediction

Looking at some examples of word pairs in the dataset, it is possible to compare the spaces behavior in the task:

Word pair	Gur350 $0 \leq sim \leq 4$	Standard $0 \leq sim \leq 1$	SVD $0 \leq sim \leq 1$	NMF $0 \leq sim \leq 1$
1. Agentur-n Irrtum-n (En: *agency-n error-n*)	0.0	0.05	0.08	0.02
2. analysieren-v Analyse-n (En: *analyse-v analysis-n*)	3.8	0.0	1e-13	0.0
3. Ansehen-n Schaden-n (En: *reputation-n damage-n*)	0.8	0.17	0.3	0.05
4. Aufstieg-n Erfolg-n (En: *promotion-n success-n*)	3.2	0.25	0.4	0.4
5. Aussage-n Rede-n (En: *statement-n speech-n*)	2.3	0.25	0.5	0.5
6. Auto-n fahren-v (En: *car-n drive-v*)	3.5	0.0	-7e-13	8e-05

Table 17: Examples of word pairs from Gur350 with values assigned by the dataset and the semantic spaces derived from the reduced DM.XI and its transformed versions

- In a case like pair 1, human and model similarities are all very close to zero. Thus, the models all make good predictions about that test pair.
- A pair like 5 which is assigned by native speakers a medium value of similarity is an example of a better result of the transformed models in comparison with the standard one, since SVD and NMF assigned the a medium value of similarity too. However, in a case like 3, the standard space predicts a value which is more similar to the average human judgement than its reduced versions, with SVD getting closer to it than NMF.
- Pairs of very similar words (human judgement: > 3), like 2 and 6 not contained in the standard model get with the transformations a zero (like in the original) or a very close to zero value, ending up decreasing strongly the value of correlation with the dataset of

reference. In case 6, SVD is even assigning a negative value: as explained in Chapter 3.1, values lower than 1 can be obtained with this factorization and it is not clear how they should be interpreted. In the same way, negative similarity measurements derived from a space transformed with SVD do not have a clear semantic explanation and are excluded in both *All* and *Covered* condition.

In summary, the experiment suggests that dimensionality reduction applied to this crosslingual distributional memory negatively affects the quality of the original model. Though size, efficiency and coverage are improved, the performances in word similarity prediction worsens. However, between the two algorithms of matrix factorization, SVD keeps giving better results than NMF on syntax-based distributional semantics models.

5. Dimensionality reduction on a multilingual distributional memory

5.1 Methodology and tools

Matrix factorization can also be used in order to obtain a multi-lingual model by merging two distributional memories, derived from corpora of different languages. Therefore, combining a DM obtained via crosslingual method and one via traditional method, the size of the resulting model can then be reduced applying transformations to it. In this way it is possible to take advantage both of the information derived from English corpora and both from the ones of the target language, just like DM.MULTI, but with a reduced unified model.

As a matter of fact, a multilingual model already exists but combines the resulting semantic similarities derived from the two models and not directly themselves. If dimensionality reduction could be able to reduce the size of the merged DM without a substantial loss in quality in comparison to DM.MULTI, it would not be necessary to rely on one model or another depending on the value of similarity assigned, like with Backoff and MaxSim, and instead using word vectors derived from a single model.

Two methods have been considered for merging one crosslingual DM and one monolingual DM, distinct by the way the two models are concatenated: in the first case they are merged as $W \times LW$ matrices, in the second one as tensors.

Depending on the way the merged distributional memory is reduced to the top link-word pairs (according to the usual procedure), the choice between these methods affects the size of the resulting model. As a matter of fact, the former implies reducing the two distributional memories independently to their respective top 60K link-word pairs and then merging them in the format of sparse matrices, whereas the second one implies doing the reduction directly on the merged distributional memories. Thus, the top 60K pairs are calculated on the two DMs together, which means that the link-word tuples considered are in this case the half of the ones with the other method, resulting in a smaller model but theoretically less balanced between the two DMs. The effect is anyway small, since a crosslingual DM like DM.Xl is much larger than a monolingual like DM.De and the difference of the amount of links coming from that is always consistently bigger than the one belonging to the other.

After the first rough context reduction, SVD and NMF can be applied using Dissect toolkit and similarity values can be computed, according to the same procedure followed for the dimensionality reduction and evaluation of DM.Xl.

5.2 Merging DM.De and DM.Xl

The methodology exposed above has been applied for merging DM.De and DM.Xl, thus two models for German, one built via traditional method (from German corpora) and one via translation (from English corpora). This has meant unifying DMs with respectively 78M and 1.7B entries. As already mentioned, the different types of merging are supposed to result in a model less balanced than the other with respect to the amount of entries coming from the crosslingual DM, but the unequal sizes of the DMs, as in this case, almost nullifies this effect. In fact, with the first method DM.De entries in the merged DM are still only 9.9% and with the second 5.7%, thus the main difference between the resulting models remains their size.

The two methods of model combination have required the following steps:

Method A

1) Dm.Xl and DM.De are both reduced on the basis of their top 60K link-word pairs (top 20K respectively for nouns, verbs and adjectives).
2) Each reduced distributional memory is turned into a sparse matrix in the `sm` format.
3) The two matrices are concatenated by adding a prefix to each link, in order to distinct the ones belonging to DM.Xl and to DM.De.
4) SVD and NMF are applied to the resulting sparse matrix by reducing the size to 500 dimensions.

Method B

1) DM.Xl and DM.De entries are directly concatenated in a new distributional memory, adding a prefix to the links according to their origin.
2) The DM obtained by merging the two models is reduced on the basis of the top 60k link-word pairs (top 20K respectively for nouns, verbs and adjectives).
3) The model is turned into a sparse matrix in `sm` format.
4) SVD and NMF are applied by reducing the size to 500 dimensions.

After having combined the models, the evaluation is done with the usual task of word similarity prediction, using the Gur350 dataset.

5.3 Results evaluation

Method A

Word similarity prediction (Gur350)					
Semantic space	*All*		*Covered*		
	Correlation		Correlation (Pearson's ρ)		Coverage
	Standard space	Dataset	Standard space	Dataset	
DM.De + DM.Xl (Method A)					
Standard	-	.33	-	.35	.61
SVD	.78	.30	.74	.29	.74
NMF	.60	.17	.48	.15	.91
DM.MULTI					
Backoff	-	.40	-	.45.	.69
MaxSim	-	.42	-	47	.69

Table 18: Correlation and coverage values in word similarities prediction of semantic spaces derived from DM.Xl + Dm.De with Method A (top 60K link-word pairs version) and its reduced versions

In the case of the model resulting from Method A, thus the one with the bigger size, coverage is increased in comparison to DM.Xl and its reduced version in all of the spaces as expected by the merging of the two models. However, performances in general also are not as good as the ones of both Backoff and MaxSim.

The standard reaches a good level of correlation though not competitive with the ones of the other DMs available for German. Transformations increase coverage, with a particularly strong rise in the case of NMF, which almost reaches the same values of baseline and word-based models[30]. SVD performances as usual are anyway better than NMF but since its similarity values are not enough strongly correlated with the standard ones,

[30] See Table 8

it is not as good as the original model.

The plots showing the correlation between respectively SVD and NMF, and the original space, and between each semantic space with the dataset, in *Covered* condition, are reported below.

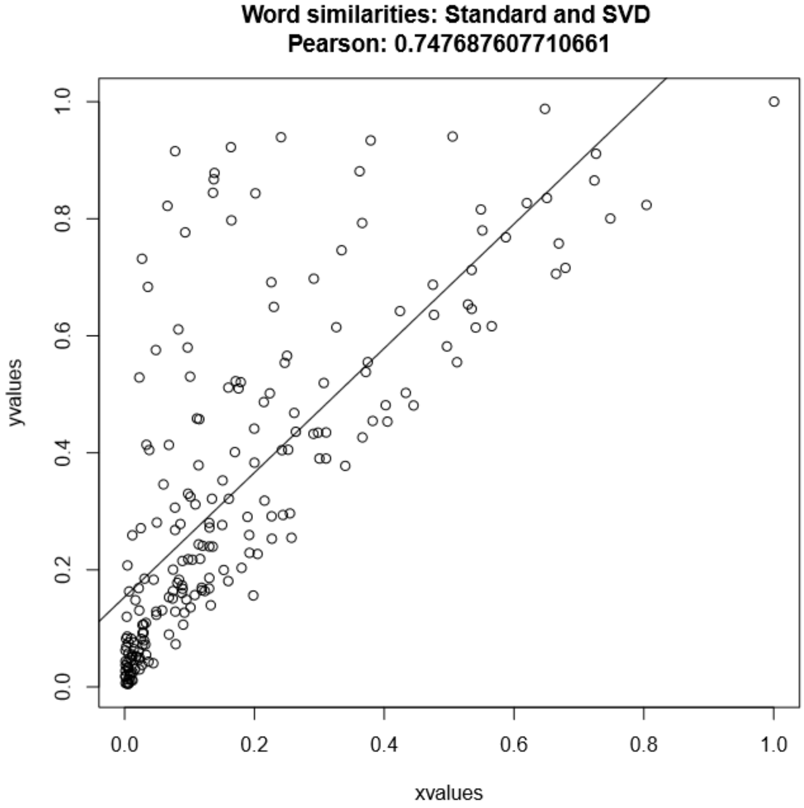

Figure 8: Diagram of the correlation between the standard version of DM.De + DM.Xl obtained with method A and its SVD version

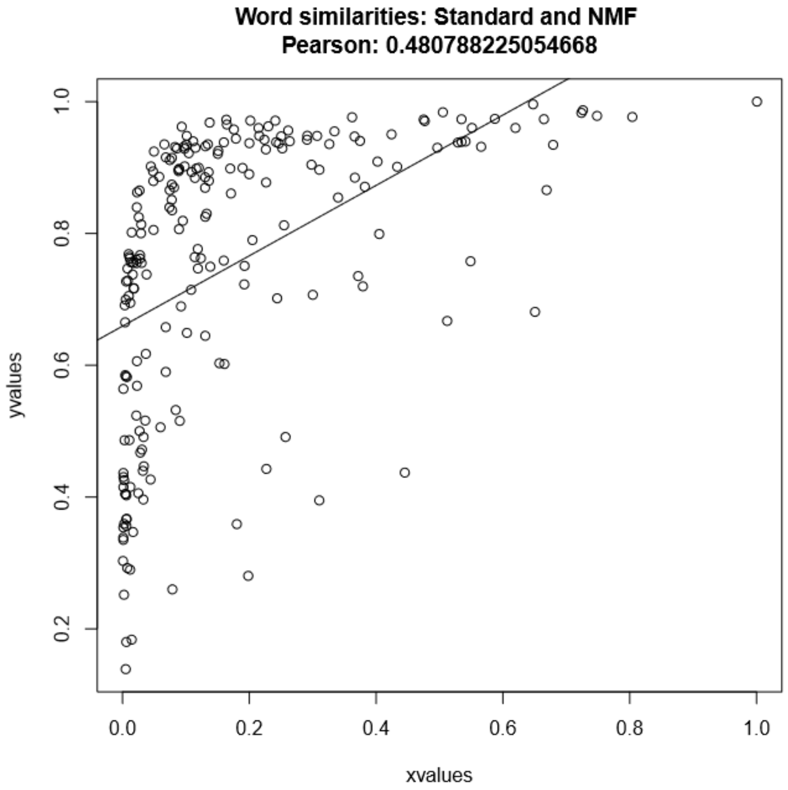

Figure 9: Diagram of the correlation between the standard version of DM.De + DM.Xl obtained with method A and its NMF version

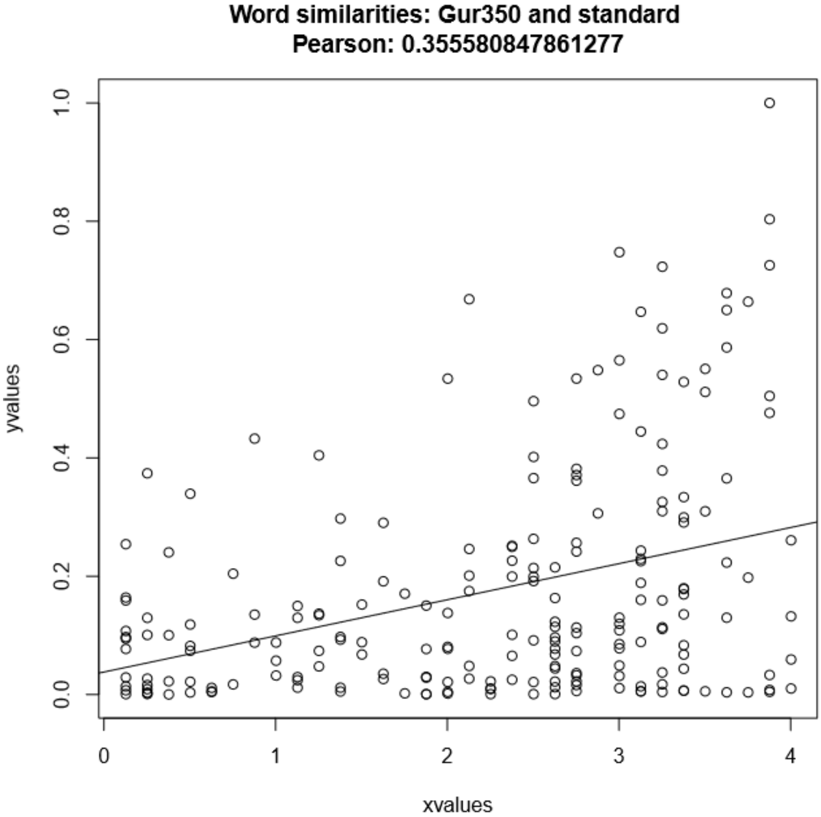

Figure 10: Diagram of the correlation between the standard version of DM.De + DM.Xl obtained with method A and Gur350 dataset in word similarity prediction

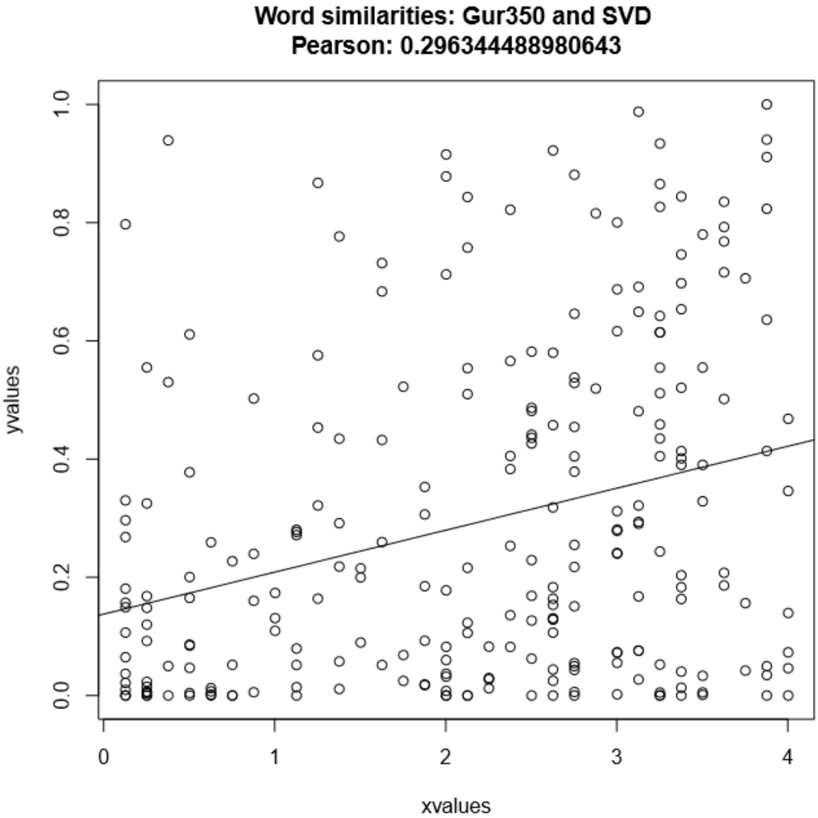

Figure 11: Diagram of the correlation between the SVD version of DM.De + DM.Xl obtained with method A and Gur350 dataset in word similarity prediction

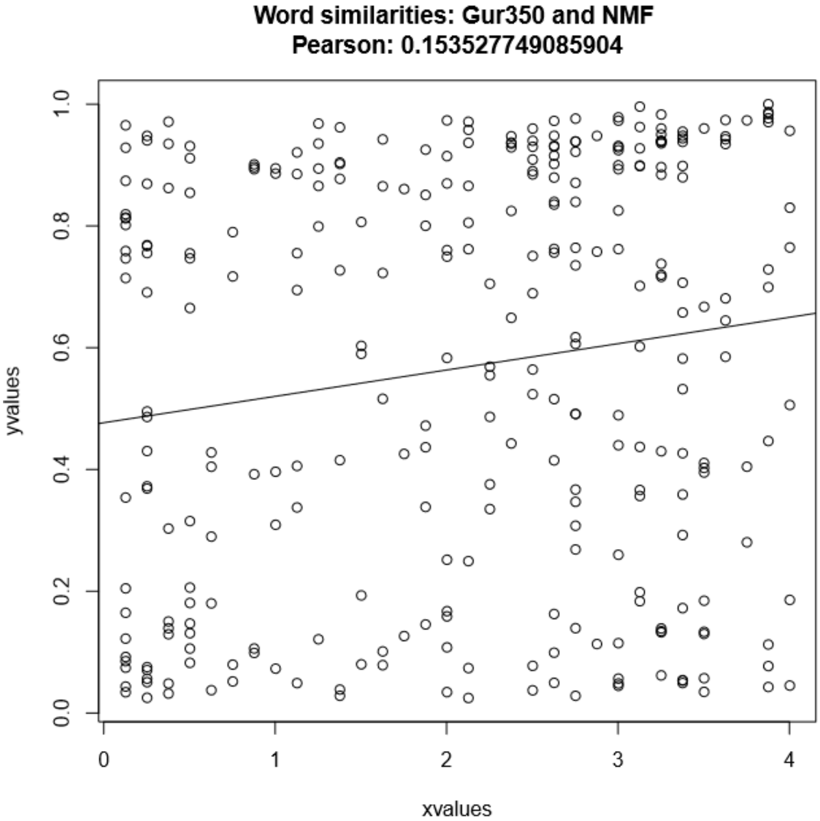

Figure 12: Diagram of the correlation between the NMF version DM.De + DM.Xl obtained with method A and Gur350 dataset in word similarity prediction

Method B

Word similarity prediction (Gur350)					
Semantic space	*All*		*Covered*		
	Correlation		Correlation (Pearson's ρ)		Coverage
	Standard space	Dataset	Standard space	Dataset	
DM.De + DM.Xl (Method B)					
Standard	-	.23	-	.30	.36
SVD	.99	.24	.98	.29	.40
NMF	.74	.18	.44	.18	.50
DM.MULTI					
Backoff	-	.40	-	.45.	.69
MaxSim	-	.42	-	47	.69

Table 19: Correlation and coverage values in word similarities prediction of semantic spaces derived from DM.Xl + Dm.De with Method B (top 60K link-word pairs version) and its reduced versions

As expected, the smaller number of link-word pairs considered in this model results in a lower coverage (the value is even lower than the other DMs available for German[31]). SVD version has a very strong correlation with the standard one; therefore their performances are very similar. The reduction implemented with NMF instead has the usual low correlation with the standard model and with the dataset. The values predicted by the standard space and SVD version come close to human judgements with a correlation of about .3, which is however inferior to the one of the other DMs.

.

[31] See Table 8

The diagrams below show the correlation between respectively SVD and NMF, and the original space, and between each semantic space with the dataset, in *Covered* condition.

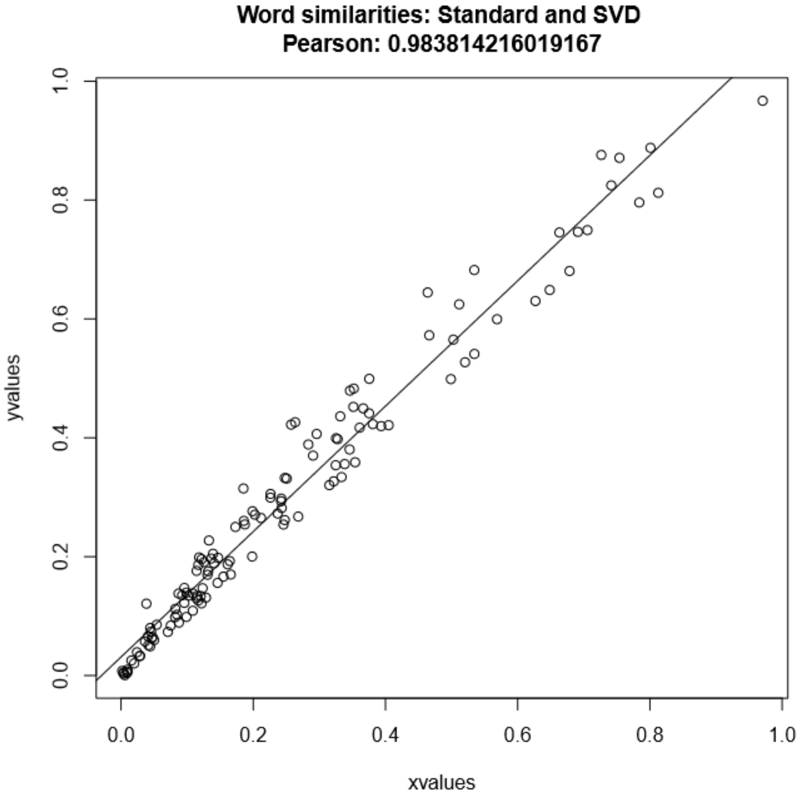

Figure 13: Diagram of the correlation between the standard version of DM.De + DM.Xl obtained with method B and its SVD version

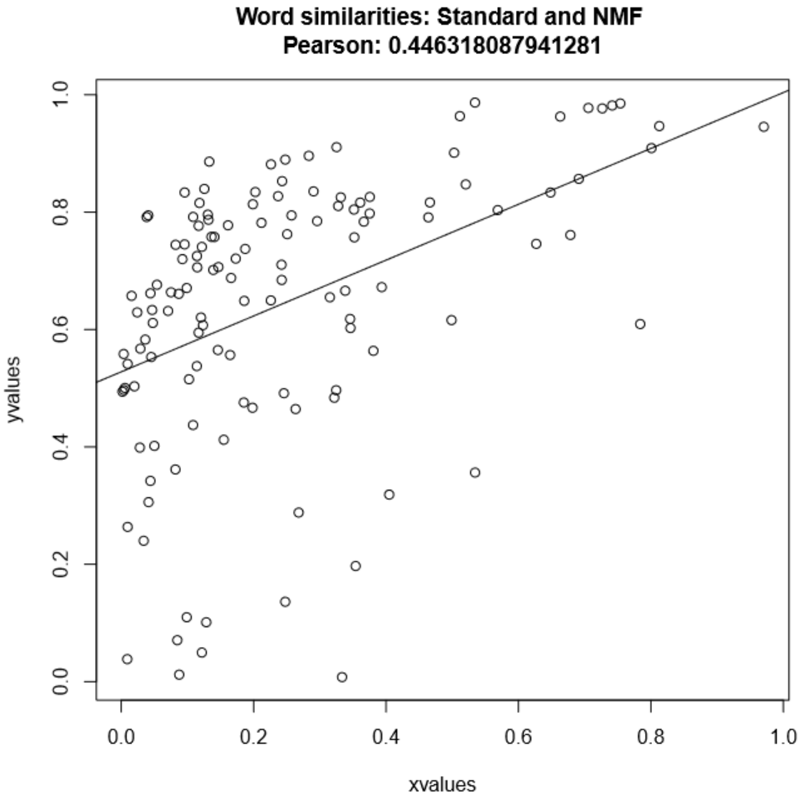

Figure 14: Diagram of the correlation between the standard version of DM.De + DM.Xl obtained with method B and its NMF version

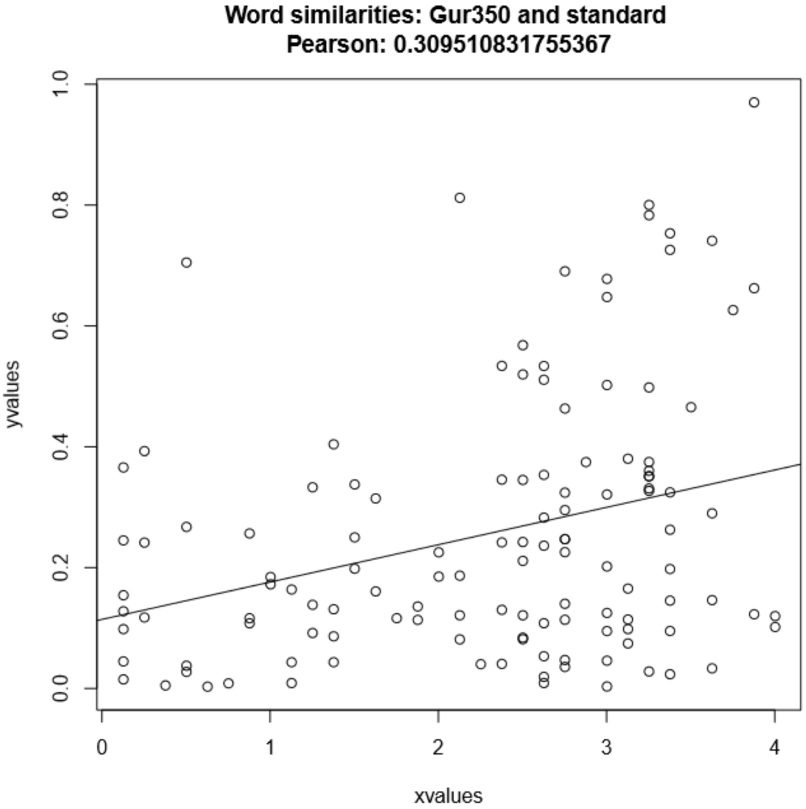

Figure 15: Diagram of the correlation between the standard version DM.De + DM.Xl obtained with method B and Gur350 dataset in word similarity prediction

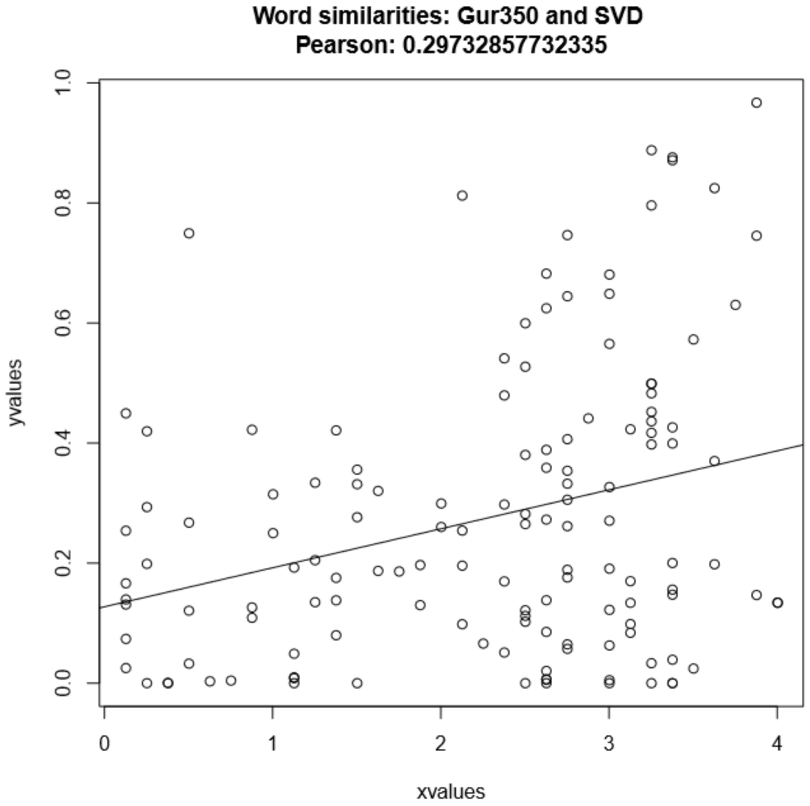

Figure 16: Diagram of the correlation between the SVD version DM.De + DM.Xl obtained with method B and Gur350 dataset in word similarity prediction

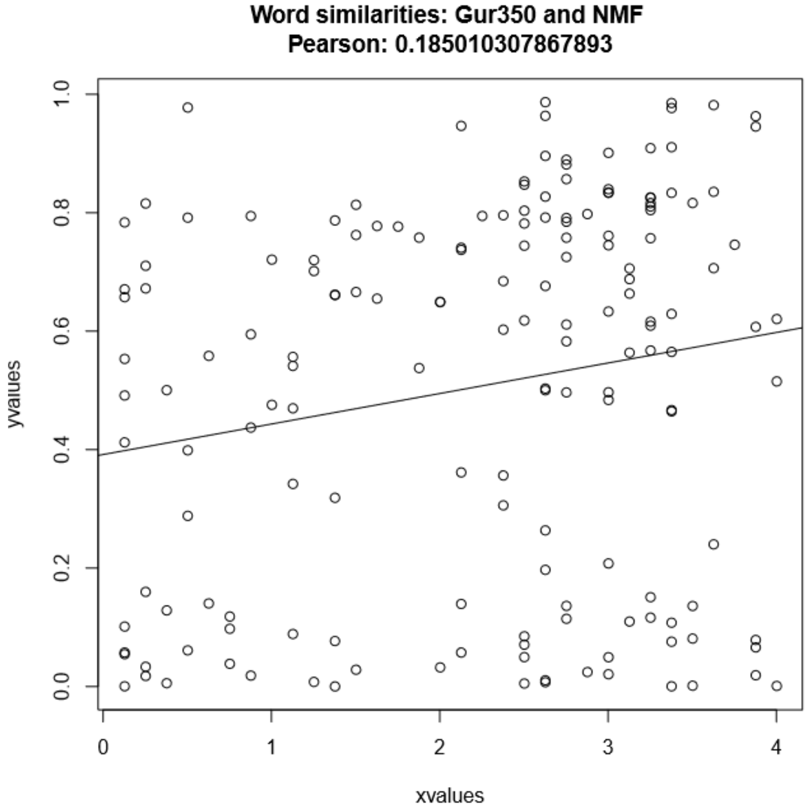

Figure 17: Diagram of the correlation between the NMF version DM.De + DM.Xl obtained with method B and Gur350 dataset in word similarity prediction

Word pair	Gur350 $0 \leq sim \leq 4$	Standard $0 \leq sim \leq 1$	SVD $0 \leq sim \leq 1$	NMF $0 \leq sim \leq 1$
1. `Agentur-n Irrtum-n` (En: *agency-n error-n*)	0.0	A: 0.02 B: 0.001	A: 0.03 B: 0.007	A: 0.50 B: 0.49
2. `analysieren-v Analyse-n` (En: *analyse-v analysis-n*)	3.8	A: 0.0 B: 0.0	A: 1e-13 B: -1e-14	A: 0.04 B: 0.01
3. `Ansehen-n Schaden-n` (En: *reputation-n damage-n*)	0.8	A: 0.08 B: 0.25	A: 0.16 B: 0.42	A: 0.89 B: 0.79
4. `Aufstieg-n Erfolg-n` (En: *promotion-n success-n*)	3.2	A: 0.11 B: 0.35	A: 0.45 B: 0.45	A: 0.93 B: 0.80
5. `Aussage-n Rede-n` (En: *statement-n speech-n*)	2.3	A: 0.25 B: 0.34	A: 0.45 B: 0.47	A: 0.94 B: 0.60
6. `Auto-n fahren-v` (En: *car-n drive-v*)	3.5	A: 0.0 B: 0.0	A: -3e-13 B: -7e-15	A: 0.03 B:0.001

Table 20: Examples of word pairs from Gur350 with values assigned by the dataset and the semantic spaces derived from DM.Xl + DM.De with both methods A and B and their transformed versions

Table 20 shows some similarity values assigned by the spaces both in the A and B models. Phenomena similar to the ones observed for the reduced version of DM.Xl occur. Moreover, it can be observed a general tendency of NMF to assign much higher numbers: with method B, this happens less strongly and this indeed corresponds to a slightly higher correlation with the dataset.

In summary, merging DM.Xl and DM.De in a multilingual model to which dimensionality reduction is then applied, does not achieve the same quality as the other methods already implemented. Therefore, DM.Multi Backoff and MaxSim both have better performances in the task of word similarity prediction in terms of correlation with the values of a dataset and of coverage[32]. Moreover, these models have been outperformed by other

[32] See Table 8

multilingual models, as well as by other DMs. However, the first merging method outperforms the second, with a higher coverage and a better correspondence with native speakers judgements.

Conclusions

The availability for a language of a syntax-based distributional semantics model with the structure of a distributional memory represents an important opportunity for investigating semantic phenomena. Translating an already existing English model of this kind into a target language overcomes the problem of the lack of corpora and parsers of a quality comparable to those in the source language, obtaining, in the case of German, a crosslingual DM that beats in quality the existing monolingual model.

Anyway, with translation the size of this type of model increases substantially and methods of data reduction seems to be necessary in order to have improvements in efficiency and performances. Dimensionality reduction is one of the possible approaches to ths issue: by applying matrix factorization, it reduces the semantic space to a much smaller number of dimensions decreasing computational costs, and discovering latent information in the model overcoming the problem of data sparseness.

Therefore, this method has been implemented during the project on the crosslingual DM available for German, testing both the algorithms of SVD and NMF type of factorization, expecting a decrease in the size of the model together with an equal or better quality than the original semantic space.

Moreover, another experiment has been carried out during the project. A multilingual DM that exploits resources both in English and in the target language, by combining a crosslingual and a monolingual model, is also an appealing semantic resource, because its coverage is higher than the one of each single model and it has complementary properties derived from each DM. One such model (DM.MULTI) has been built for German by combining resulting similarities, but dimensionality reduction can also be another way to create a model of this kind with a still manageable size, by merging the two original models and then applying matrix factorization. During this project, SVD and NMF have been applied to the DM derived by unifying the crosslingual and the monolingual models available for German.

All these reduced DMs have been evaluated in the task of word similarity prediction by comparing values assigned by the model with the Gur350 collection of human relatedness judgments.

The potentiality of dimensionality reduction to reduce data without loss in quality obtaining at the same time an improvement thanks to the ability to generalize over data has revealed in both the two experiments not to be sufficiently satisfactory when applied to this type of DM. Size reduction results in a worsening in performances and the models behaviors are often unpredictable, especially using NMF, as models respond differently to this transformations. Though some generalizations can be made:

- SVD transformation gives almost always better results than NMF. Its behavior is more predictable and it is usually strictly correlated with the original model.
- NFM transforms the standard semantic space in a much deeper way than SVD and its value of correlation with the original is usually lower. This often does not get closer to human judgments and instead performances in word similarity prediction are worse.
- In the case of the crosslingual DM, SVD and NMF versions of the space, whose results do not reach levels comparable with the ones available for German, are almost always outperformed by the standard space . This corresponds to the matrix derived from the original DM reduced to a certain number of top link-word pairs. This first reduction does not seem to cause the loss of information, thus it is possible to entail that a reduction of this model by using a filter of this kind may also be another possible approach to the problem of size, though its simplicity.

- Matrix transformation used for merging two German DMs is not a viable approach to the building of a multilingual model as results are worse than with the other already existing DMs. Anyway, merging the two DMs by applying independently to each one a first reduction to the top link-word pairs and then concatenating them in the form of a sparse matrix has proved to be a better method than merging them at the beginning and then filtering out the less relevant contexts.

One of the possible directions for further research is then, a more detailed analysis of dimensionality reduction operations applied on distributional semantics models, in order to find out some patterns of behavior that may help predicting how a model of this type may be affected by the transformation. As a matter of fact, though some conclusions can be drawn from the experiments, a more in-depth look at the math and the algorithm behind the transformation and the response of the space in details may better explains the reasons of such results.

On the other side, testing other possible methods of size reduction different from this, both on the crosslingual DM and the combination of a monolingual and a crosslingual one, is another track that could be worth being pursued. A more efficient data structure like Bloom filter or other type of transformations on the structure of the data, like tensor or graph sparsification may be tried out. Moreover the good results obtained by reducing the space to a number of top link-word pairs suggest that refining this method to decrease data size could also be a possible approach to the issue.

Bibliography

Marco Baroni, Alessandro Lenci, *Distributional Memory: A general Framework of Corpus-Based Semantics*, Computational linguistics, n. 4, p. 1-49, vol. 36, 2010

Marco Baroni, Eduard Barbu, Brian Murphy, Massimo Poesio, *Strudel: A distributional semantics model based on properties and types*, Cognitive Science, 34(2):222–254, 2010

Georgiana Dinu, Nghia The Pham, Marco Baroni, *DISSECT- DIStributional Semantics Composition Toolkit*, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, p. 31–36, 2013

Zellig Harris, *Distributional structure,* Word, 10(2-3), p. 1456–1162, 1954

Thomas K. Landauer, Susan T. Dumais, *A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge*, Psychology Review, 104:211–240, 1997

Daniel D. Lee and H. Sebastian Seung, *Algorithms for non-negative matrix factorization*, NIPS, p. 556–562, 2000

George Miller, Charles Walter, *Contextual correlates of semantic similarity,* Language and Cognitive Processes, 6:1–28, 1991

Saif Mohammad, Iryna Gurevych, Graeme Hirst, Torsten Zesch, *Cross-Lingual Distributional Profiles of Concepts for Measuring Semantic Distance*, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), p. 571–580, Prague, Czech Republic, 2007

Gregory Murphy, *The Big Book of Concepts*, MIT Press, Cambridge, MA, 2002

Sebastian Padò, Mirella Lapata, *Dependency-based construction of semantic space models*, Computational Linguistics, 33(2):161–199, 2007

Sebastian Padò, Jason Utt, *Crosslingual and Multilingual Construction of Syntax-Based Vector Space Models*, Transactions of the Association of Computational Linguistics, 2014

Sebastian Padò, Jason Utt, *A Distributional Memory for German*, Proceedings of KONVENS 2012, LexSem 2012 workshop, p. 462—470, ed. Jeremy Jancsary, 2012

Timothy Rogers, James McClelland, *Semantic Cognition: A Parallel Distributed Processing Approach*, MIT Press, Cambridge, MA, 2004

Peter D. Turney, Patrick Pantel, *From Frequency to Meaning: Vector Space Models of Semantics*, Journal of Artificial Intelligence Research, p. 141-188, vol.37, 2010

Tim Van de Cruys, *Mining for Meaning. The Extraction of Lexico-semantic Knowledge from Text*, PhD thesis, University of Groningen, The Netherlands, 2010

Sitography

DISSECT 0.1.0 documentation,
`http://clic.cimec.unitn.it/composes/toolkit/`
visited on 23rd March 2015

Gensim library
`https://radimrehurek.com/gensim/`
visited on 28th March 2015

German relatedness datasets
`https://www.ukp.tu-darmstadt.de/data/semantic-relatedness/german-relatedness-datasets/`
visited on 26th March 2015

The MEN Test Collection
`http://clic.cimec.unitn.it/~elia.bruni/MEN.html`
visited on 23rd March 2015

The WordSimilarity-353 Test Collection
`http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/`
visited on 23rd March 2015

Acknowledgments

A special thanks goes to the Institüt für Maschinelle Sprachverarbeitung (IMS) of the University of Stuttgart for having given me the precious opportunity of working at this project in such a stimulating and welcoming environment, and consequently of growing as a student as well as a human being. In particular, I am very grateful to Prof. Sebastian Padò and Jason Utt.

My gratitude is also for Prof. Alessandro Lenci for having supported me during this whole experience, from the beginning up to the writing of this thesis.

Thanks to my family, that loving me has pushed me to believe in my capabilities. I am grateful to my parents, who have taught me how to pose questions to myself and how to be eager to look for answers, and my sister, together with whom I have grown up learning how to always keep our minds curious.

I owe a debt of gratitude to my friends, as they helped me both to understand more my brain and also to turn it off when necessary.

Finally, I want to thank whoever has criticized me at least once for giving too much value to the meaning of words, or trying to “factorize” reality too much: now I have learnt that I can turn a potential flaw into something useful.