A Study on the Interplay Between the Corpus Size and Parameters of a Distributional Model for Term Classification

Behrang Q. Zadeh
Heinrich-Heine-Universität Düsseldorf
zadeh@phil.hhu.de

Abstract

We propose and evaluate a method for identifying co-hyponym lexical units in a terminological resource. The principles of term recognition and distributional semantics are combined to extract terms from a similar category of concept. Given a set of candidate terms, random projections are employed to represent them as low-dimensional vectors. These vectors are derived automatically from the frequency of the co-occurrences of the candidate terms and words that appear within windows of text in their proximity (context-windows). In a k-nearest neighbours framework, these vectors are classified using a small set of manually annotated terms which exemplify concept categories. We then investigate the interplay between the size of the corpus that is used for collecting the co-occurrences and a number of factors that play roles in the performance of the proposed method: the configuration of context-windows for collecting co-occurrences, the selection of neighbourhood size (k), and the choice of similarity metric.

1 Introduction

Automatic term recognition (ATR) deals with the extraction of domain-specific lexical units from text. The input of ATR is a large collection of documents, i.e., a *special corpus*,[1] and the output is a vocabulary that is used for communicating specialized knowledge (L'Homme, 2014). This vocabulary comprises a collection of single-token and multi-token lexical units—respectively known as *simple* and *complex* terms—that form a terminological resource. For example, in computational linguistics, *lexicon* and *parsing* are examples of simple terms, while *multilingual corpus* and *information extraction* are complex terms. Similarly, in molecular biology, *collagen* and *cortisol* are examples of simple terms, and *I kappa B* and *plasma prednisolone* are examples of complex terms.

Terms, extracted by an ATR system, represent a broad spectrum of concepts that exist in a domain knowledge. Terms and their corresponding concepts, however, can be further organized in several categories to form a taxonomy; each category characterizes a group of terms from 'similar' concepts in the domain of study (Figure 1). For example, in computational linguistics, the terms *lexicon* and *multilingual corpus* can be categorized under the category of *language resources*, while *parsing* and *information extraction* can be categorized under the concept of *technologies*. Likewise, in molecular biology, instances such as *collagen* and *I kappa B* are categorized as *proteins*, while *cortisol* and *plasma prednisolone* are classified as *lipid substances*.

If the concept categories are not known, a method is used to suggest an organization for terms (e.g., Dupuch et al. (2014)); Cederberg and Widdows (2003)). However, concept categories are usually known, or at least, a partial knowledge of them exists. In these scenarios, typically a manually annotated corpus is employed to develop an entity tagger in a supervised fashion, often in the form of a sequence classifier. Bio-entity tagging is an established example of this kind of tasks (Nobata et al., 1999). These methods, however, rely heavily on manually annotated corpora, in which each mention of a term and its concept-

[1] Following the terminology proposed by Sinclair (1996), we use the term special corpus; that is, a corpus containing sublanguage material.

Proceedings of the 5th International Workshop on Computational Terminology,
pages 62–72, Osaka, Japan, December 12 2016.

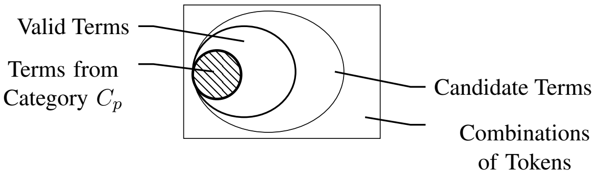


Figure 1: Venn diagram that illustrates the relationship between candidate terms, valid terms, and a particular category of terms C_p. ATR targets the extraction of candidate terms and the identification of valid terms. However, term classification targets the identification of terms that belong to a concepts-category, i.e., a subset of valid terms.

category must be annotated. Provided that enough training data is available, a reasonable performance can be attained in these recognition tasks (Kim et al., 2004).

Yet in several scenarios, the targeted concept categories (similar to entity recognition tasks) are known but no manual annotation is available for the training and development of an entity tagger. This is a familiar problem when a terminological resource with a hierarchical structure must be constructed from scratch, a task with many practical applications (see e.g. Chakraborty et al. (2014)) and renewed interests, e.g., as addressed in *cold-start knowledge base population* (Ellis et al., 2012; Mayfield et al., 2014) and ontology learning. Similarly, this problem surfaces in maintaining terminologies, where constant update and extension is required to accommodate new vocabularies and their usages (Habert et al., 1998).

This paper suggests a method to address this situation: the extraction of terms from a particular class of concepts in the absence of training data for the development of an entity tagger. The proposed method (similar to ATR and in contrast to entity recognition task) works at the corpus level and does not deal with individual term mentions. However, in contrast to ATR (which extracts terms from diverse concept categories in a specific domain knowledge) and similar to entity tagging, the proposed method is designed to extract a subset of terms that belongs to a particular category of concepts in a domain knowledge (i.e., *co-hyponym* terms). Note that each category can be further organised into more refined subcategories to provide abstractions at different levels of granularity. Since co-hyponymy is an inheritable relationship, terms under each category, disregarding the subcategory that they belong to, are still co-hyponym.

Since polysemy is less frequent in specialized vocabularies than in general vocabularies, the proposed approach is effective and useful. We support this claim with a comparison between the proportion of polysemous entries in WordNet (Miller, 1995), i.e., a general vocabulary, and the terminological resource that is induced from the annotated terms in the GENIA corpus (Kim et al., 2003). In WordNet, approximately 17% of entries are polysemous. The GENIA corpus (which is a well-known special corpus in the domain of molecular biology) provides manual concept-category annotations for 92,722 term mentions. These term mentions constitute a vocabulary of 34,077 distinct entries, of which only 1,373 are polysemous (i.e., their individual mentions are annotated with at least two concept categories). Therefore, compared to WordNet, the GENIA terminological resource contains only a small fraction of polysemous entries, i.e., $\frac{1372}{34077} = 4\%$.[2]

The proposed term classification method is realized as an ad hoc term-weighting procedure on top of an ATR system. ATR typically comprises a two-step procedure: candidate term extraction followed by term weighting and ranking (Nakagawa and Mori, 2002). Candidate term extraction deals with term formation and the extraction of candidate terms (Ananiadou, 1994). Following the extraction of candidate terms, as stated byKageura and Umino (1996), an ATR system often combines scores that are known as *unithood* and *termhood* to weight terms. Unithood indicates the degree to which a sequence of tokens can be combined to form a complex term. It characterizes syntagmatic relations between tokens to identify collocations (therefore is only defined for complex terms). Termhood, however, “is the degree that a linguistic unit is related to $\cdots$ some domain-specific concepts” (Kageura and Umino, 1996). Hence, termhood is defined for both simple and complex terms. From a linguistic perspective, termhood char-

[2] This comparison can be biased since WordNet has been designed and developed to provide a comprehensive picture of words and their meanings. Therefore, the proportion of polysemous words in a reference corpus (as defined in Sinclair (1996)) can be less than %17. Still, we maintain polysemy is far more frequent in reference corpora than in special corpora.

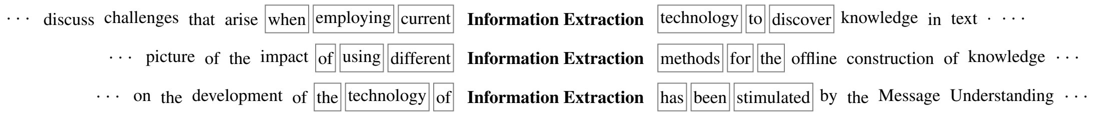

··· discuss challenges that arise when employing current Information Extraction technology to discover knowledge in text ····
··· picture of the impact of using different Information Extraction methods for the offline construction of knowledge ···
··· on the development of the technology of Information Extraction has been stimulated by the Message Understanding ···

Figure 2: Shown a context-window of size 3 tokens that extend around terms: the occurrences of the candidate term *information extraction* in different sentences of a corpus. For each occurrence of the candidate term in each line, the context-window consists of words that are placed in rectangles. To construct a model, these co-occurrences are collected, counted, and represented by a vector.

acterizes an associative relationship between terms and the communicative context that verbalizes their meaning (in this scenario, the corpus). The major difference between the proposed term classification technique and a general ATR system is, therefore, the way they define termhood.

To actualize the proposed term classification task, a termhood measure that can identify co-hyponym terms must be devised. To achieve this goal, we take a distributional approach. We assume that the association of a term to a concept category is a kind of relation that can be modelled using the syntagmatic relation of the term and its co-occurred words in context-windows extended in the vicinity of the term's mentions in the corpus (Figure 2). We, therefore, hypothesise that co-hyponym terms tend to have similar distributional properties in these context-windows. Note that a similar hypothesis has been employed in many other distributional techniques for terminology extraction. In order to quantify these distributional similarities, vector space models are employed (Turney and Pantel, 2010).

Words that appear in context-windows are represented by the elements of the standard basis of a vector space (i.e., informally each dimension of a vector space) and each candidate term is represented by a vector. In this vector space, the co-occurrence frequency of words and candidate term in context-windows determines the coordinates of the vector that represent the candidate term. Hence, the values assigned to the the vector's coordinates represent the correlation between the candidate term that the vector represents and the words in context-windows. Consequently, we can use the proximity of candidate terms to compare their distributional similarities in this *term-space model*.

In this term-space model, we model a category of terms using a set of *reference terms* (shown by R_s), i.e., a small number of terms that are manually annotated with their corresponding concept category. The averaged distance between vectors that represent candidate terms and the vectors that represent R_s is assumed to determine the association of candidate terms to the concept categories represented by R_s. This association is computed using a k-nearest neighbours (k-nn) method. As explained by Daelemans and Van Den Bosch (2010), the memory-based k-nn technique provides us with a *similarity-based reasoning framework* that can be used to identify term categories without the need for formulating these associations using a meta-language such as rules.

Like other distributional methods, finding a configuration of context-windows (i.e., the way co-occurrence frequencies are collected) that best characterizes co-hyponym terms is a major research concern that must be investigated empirically. Context-windows can be configured differently regarding the position of the candidate terms in them and the direction in which they are stretched. They can be expanded (a) only to the left side of a candidate term to collect the co-occurrences of the candidate term with preceding words in each sentence of the corpus, (b) to the right side to collect co-occurrences with the succeeding words, or (c) around the candidate term, i.e., in both left and right directions. The size of context-windows must also be decided, i.e., the extent of the region on either side of a term for collecting and counting its co-occurrences with neighbouring words. In addition, information about the order of words in context-windows can be ignored or encoded in the constructed distributional model.

Independent of the configuration of context-windows in the proposed method, due to the *Zipfian distribution* of terms and words in context-windows, vectors that represent candidate terms are inevitably high-dimensional and sparse (i.e., most of the elements of vectors are zero). The high-dimensionality of vectors hinders the computation of similarities, and their sparseness is likely to diminish the discriminatory power of the constructed model (i.e., the *curse of dimensionality* problem). To avoid these problems, a dimensionality reduction technique is employed to reduce the dimension of vectors to a certain size.

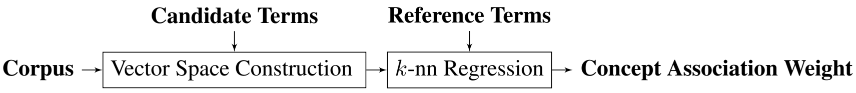


Figure 3: Method of measuring the candidate terms' association to a concept category.

Now that the vectors' dimension is set to a constant size, it is hypothesised that enlarging the size of the corpus reduces the number of zero elements in the vectors, and thus, the performance of the distributional model improves (e.g., as proposed for general language in Bullinaria and Levy (2007), Pantel et al. (2009)and Gorman and Curran (2006)). In this paper, we investigate the interplay between the size of the corpus and choosing the most discriminating configuration for context-windows in the proposed term classification task. We are interested to know (a) whether increasing the size of the corpus that is used for collecting co-occurrence frequencies enhances the performance of the classification task and (b) how doing so influences the choices that are made for configuring context-windows. Section 2 delineates the employed method. Section 3 describes the evaluation materials and framework. Results are reported in Section 4, followed by a conclusion in Section 5.

2 Method

Figure 3 illustrates the method. It is assumed that an ATR system extracts a list of candidate terms and, perhaps, ranks them by its own weighting mechanism. The extracted list of candidate terms is then processed for the construction of a vector space by scanning an input corpus. We assume that a small number of these candidate terms, e.g., 100, are annotated with their concept categories. Vectors that represent these annotated terms form a set of reference vectors R_s. In the constructed vector space, using a k-nn regression algorithm, R_s is employed to assign a concept-association weight c_w to the remaining candidate terms.

Accordingly, for a given candidate term that is represented by the vector $\vec{v}$, c_w is computed using

$$c_w(\vec{v}) = \sum_{i=1}^{k} s(\vec{v}, \vec{r_i})\delta(\vec{r_i}), \tag{1}$$

where $s(\vec{v}, \vec{r})$ denotes similarity between $\vec{v}$ and $\vec{r} \in R_s$, in which R_s is sorted by $s(\vec{v}, \vec{r})$ in descending order. If $\vec{r}$ represents a term from the targeted category of concepts, then $\delta(\vec{r}) = 1$, otherwise $\delta(\vec{r}) = 0$. While s can be defined in a number of ways, we employ three widely used definitions:

- $s(\vec{v}, \vec{r}) = \cos(\vec{v}, \vec{r})$, i.e., the cosine of the angles between $\vec{v}$ and $\vec{r}$;
- $s(\vec{v}, \vec{r}) = \frac{1}{1+\ell_2}$, where ℓ_2 is the Euclidean distance between $\vec{v}$ and $\vec{r}$; and
- $s(\vec{v}, \vec{r}) = \frac{1}{1+\ell_1}$, where ℓ_1 is the City block distance between $\vec{v}$ and $\vec{r}$.

Vector space construction is performed using *sparse stable random projections* (Li, 2007), which is implemented in the form of a sequential algorithm. Each candidate term is assigned to an m-dimensional *term vector* $\vec{t}$. Term vectors are initially empty, i.e., all the elements of $\vec{t}$ are set to zero. The input corpus is then scanned for the occurrences of candidate terms and finding their co-occurring words in context-windows (e.g., see Figure 2). Each of these words is assigned exactly to one *word vector* $\vec{w}$. Similar to term vectors, word vectors are also m-dimensional. However, the elements w_j of each $\vec{w}$ are instantiated with random values with the following distributions:

$$w_j = \begin{cases} \lfloor \frac{-1}{U_1} \rfloor & \text{with probability } \frac{1}{2\alpha} \\ 0 & \text{with probability } 1 - \frac{1}{\alpha} \\ \lfloor \frac{1}{U_2} \rfloor & \text{with probability } \frac{1}{2\alpha} \end{cases}. \tag{2}$$

Once a $\vec{w}$ is generated and assigned to a word, it is stored and kept for later usages.

If the similarity between $\vec{v}$ and $\vec{r}$ is measured using the cosine or Euclidean distance (i.e., in an ℓ_2-normed space), then U_1 and U_2 are set to 1 and $\alpha = \mathrm{O}(\sqrt{|\vec{w}|})$, where $|\vec{w}|$ is the number of word vectors.

$\mathbf{T_{Mention}}$	$\mathbf{P_{Mention}}$	$\mathbf{T_{Distinct}}$	$\mathbf{P_{Distinct}}$	$\mathbf{T_{Polysemy}}$	$\mathbf{P_{Polysemy}}$
92,722	34,264	34,077	8,900	1,373	403

Table 1: Statistics of the terminological resource: terms and 'protein terms' are respectively abbreviated by T and P (note P $\subset$ T).

In this case, $\vec{w}$ vectors resemble a random projection matrix with asymptotic Gaussian distribution. However, if the similarities are measured using the city block distance (i.e., in an ℓ_1-normed space), then U_1 and U_2 are two independent uniform random variables in $(0, 1)$ and $\alpha = \mathrm{O}(\sqrt{|\vec{w}|/100})$, where $|\vec{w}|$ is the number of word vectors and the constant factor 0.01 is an approximation of the sparsity of term-word co-occurrences in the corpus. In this case, $\vec{w}$ vectors resemble a random projection matrix with a asymptotic Cauchy distribution. Since $|\vec{w}|$ is very large, α is also relatively large; thus, the generated word vectors are highly sparse, i.e., most elements of $\vec{w}$ are set to zero and only a few have a non-zero value. To capture the co-occurrence of a candidate term and a word, the term vector $\vec{v}$ that represents the candidate term is accumulated by the word vector $\vec{w}$ that represents the word—i.e., $\vec{v} = \vec{v} + \vec{w}$. This procedure is repeated to capture all the co-occurrences of candidate terms and words that appear in context-windows in the input corpus. The result is a vector space that reflects the observed co-occurrences of terms and words at the reduced dimension m.

Subsequent to the construction of a vector space using the method described above, the distances/similarities between vectors are computed. In the ℓ_2-normed constructed vector spaces, for the given two m-dimensional vectors $\vec{v}$ and $\vec{u}$, the cosine between them is calculated using: $cos(\vec{v}, \vec{u}) = \frac{\sum_{i=1}^{m} v_i \times u_i}{\sum_{i=1}^{m} v_i^2 \times \sum_{i=1}^{m} u_i^2}$. Similarly, the Euclidean distance is given by $d_2(\vec{v}, \vec{u}) = \sqrt{\sum_{i=1}^{m} (v_i - u_i)^2}$. In the ℓ_2-normed spaces, therefore, the proposed method is equivalent to the random indexing technique (Sahlgren, 2005; QasemiZadeh and Handschuh, 2015). In the ℓ_1-normed spaces, the city block distance, however, is computed using the non-linear estimator

$$d_1(\vec{v}, \vec{u}) = \sum_{i=1, v_i \neq u_i}^{m} \ln(|v_i - u_i|).$$

In this case, the method is equivalent to the one proposed by Zadeh and Handschuh (2014). Once computed, these similarity measures are used to weight candidate terms according to Equation 1.

3 Evaluation Materials and Parameters

The proposed method is evaluated using the GENIA terminological resource. Manually annotated term mentions from the GENIA corpus (Version 3.02) are collected to build a terminological resource. This resource's entries are distinct pairs of lexical units and their annotations. The annotations are employed to organize terms in a taxonomy similar to the one proposed by Kim et al. (2004) for evaluating bio-entity taggers. To keep the reports to a manageable size, we limit the evaluation task to the identification of terms belonging to the category of *proteins* (see Table 1).

Using the the obtained frequencies in the GENIA corps and c-value measure (i.e., a widely used method for ranking terms in ATR systems (Frantzi et al., 1998)) terms are ranked in a list. From this sorted list, the top 100 terms and their annotations are used to form a set of reference vectors (R_s). Consequently, in our evaluations, R_s contains 36 *protein* terms: terms that are annotated as co-hyponyms under the concept category of 'protein' from the GENIA Ontology. Figure 4 shows the distribution of protein terms in the obtained sorted list of terms using the c-value measure with respect to a random baseline. Except for a small number of terms at the top of the list, the proportion of protein terms in the c-value sorted list is similar to the random baseline. We use the c-value ranking as one baseline in our evaluations.

To show that R_s is not sufficient for developing an entity tagger, we verify the performance of a bio-entity tagger when the employed R_s is used for its training. Namely, we employ the ABNER system, an entity tagger designed for analysing biology text (Settles, 2005). It uses conditional random fields

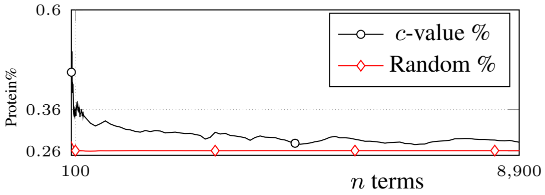


Figure 4: Proportion of protein terms in the top 8,900 terms, from lists of candidate terms sorted by the c-value measure and randomly.

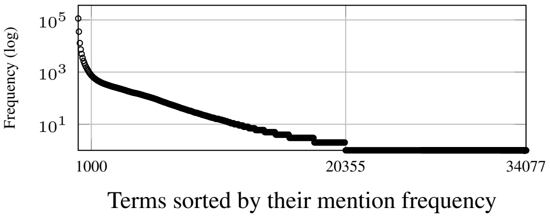


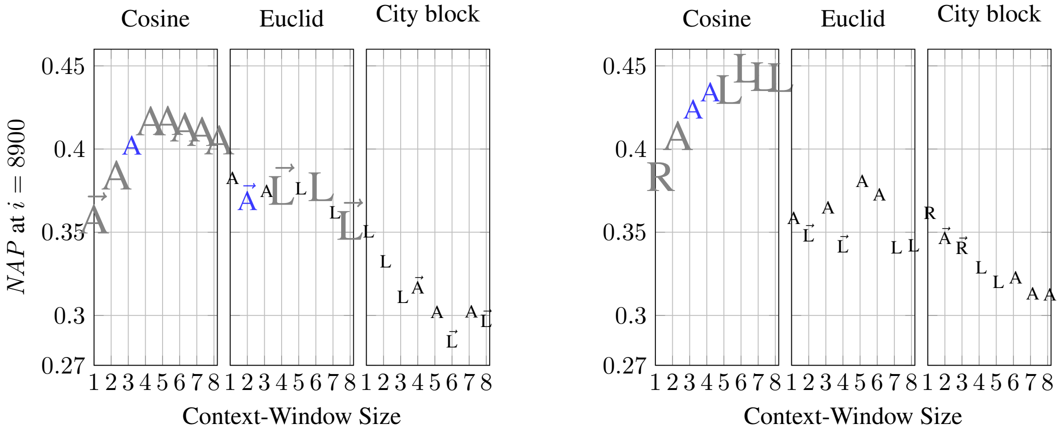
Figure 5: The frequency of terms in G_e.

and a variety of orthographic and contextual features to perform its task. If ABNER is trained using all the provided annotations for protein term mentions in the GENIA corpus, it achieves a reasonable performance (recall of 77.8 and precision of 68.1). However, if it is trained using the mentions of terms in R_s, the resulting model can only identify an additional 16 protein terms out of the remaining 8,864 terms. Put simply, the 1,321 mentions of the 36 protein terms in R_s are not sufficient to train ABNER.

Initially, we will construct vector spaces using the raw text from the GENIA corpus. Besides normalising text to lower-case letters and a simple Penn Treebank tokenisation, no other text pre-processing is performed. This pre-processing results in 490,941 tokens and a vocabulary size of 19,576. We then enlarge the corpus by fetching 223,316 abstracts from the PubMed repository, of which each abstract contains at least three of the terms in the terminological resource. The enlarged corpus has more than 55 million tokens and a vocabulary of size 881,040. Hereafter, we denote these two corpora by G_o (for the original GENIA corpus) and G_e (for the enlarged corpus). In this corpus, the terms employed in our experiments are mentioned more than 9 million times. As expected, only a small number of terms are frequent and the majority of terms are mentioned a few times. A large number of terms (i.e., about 40%) never appear in G_e (see Figure 5).

Using the method explained in Section 2, we use these two corpora to collect the co-occurrences and build vector space models. We perform our experiments with vector spaces that are constructed at the reduced dimension $m = 2000$. Considering the number of term vectors in the model (i.e., 34077), $m = 2000$ is a conservative choice that guarantees a small distortion in pair-wise distances between vectors. Similarly, because the vocabulary size $|\vec{w}| \geq 19576$, we use word vectors of 6 non-zero elements and 30 non-zero elements, respectively, for the construction ℓ_2 and ℓ_1-normed spaces. These values for the numbers of non-zero elements in word vectors are conservative choices that meet the criteria specified in Section 2 for the value of α in Equation 2.

The construction of vector spaces is carried out by collecting co-occurrence frequencies in context-windows that are configured differently regarding the direction and size in which they are stretched. Moreover, we investigate the influence of the inclusion of word order information in the model using the permutation technique described in Sahlgren et al. (2008). As suggested in research reports (see, e.g., Baroni et al. (2014) and Agirre et al. (2009)), narrow context-windows are more suitable to capture paradigmatic relations such as the one intended in this paper. Accordingly, we report the performance of the method for context-windows of $1 \leq$ size ≤ 8 tokens, for three directions of around (hereafter, denoted by A), only to the left (denoted by L), or to the right (denoted by R) of candidate terms. In addition, we construct vector spaces that encode information about the order of words in these context-windows. Hence, for each input corpus, 48 vector spaces are constructed to reflect each of the possible



(a) Obtained performances from G_o

(b) Obtained performances from G_e

Figure 6: The y-axis shows the observed NAP_i for $i = 8{,}900$ (i.e., recall 100%). For each of the employed similarity measures, the x-axis shows the size of context windows. The letters A, L, and R denote the direction in which context-windows are stretched (i.e., respectively, Around, Left, or Right side of the candidate terms). Models that encode word order information are denoted using the $\vec{\Box}$ on top. The size of letters, however, shows the value of k. The smallest size denotes $k = 1$ (black colour), while the largest size denotes $k = 25$ (grey colour); the medium size represents $k = 7$ (blue colour). In these experiments, the computed NAP over c-value ranked terms (i.e., the baseline) is 0.27. For the sake of readability, for each configuration of context-windows size and the employed similarity metric, we plot only the best observed results (complete plots are provided as supplementary materials).

configurations of context-windows, listed above.

The performance of the proposed k-nn technique is affected by the value of k. In the absence of a large training dataset, in the employed memory-based learning framework, a small value for k may lead to over-fitting and sensitivity to noise, while a large neighborhood estimation may reduce the discriminatory power of the classifier. Therefore, we report the performance of the method for three values of neighborhood size, i.e., $k \in \{1, 7, 25\}$. As stated earlier, term weighting in Equation 1 is performed by the help of three different measures: the cosine similarity, the Euclidean, and the city block distance.

4 Results

Following Schone and Jurafsky (2001), performance is measured and reported using the non-interpolated average precision at i:

$$NAP_i = \frac{1}{i}\sum_{n=1}^{i} P^n,$$

where P^n is the observed precision for extracting n protein terms. Figure 6 plots the performances that are measured by computing NAP at $i = 8900$ (i.e., 100% recall) in the obtained sets of terms that are ranked by the computed w_a (one for each of the constructed models). Independently of the size of the input corpus, the cosine similarity outperforms the Euclidean and city block distance. When the co-occurrence frequencies are collected from G_o, the best performance is obtained by using $k = 25$, in models that are built by collecting co-occurrence frequencies in context-windows of size 4 or 5 words that extend around terms. However, in experiments performed over G_e, using context-windows that expand to the left side of the candidate terms slightly outperform models that are built using context-windows that expand around the terms. As shown in Figure 6, encoding the word order information in context-windows often does not improve the performance.

Figure 7 plots the changes that are observed by enlarging the size of the input corpus. As shown, when the corpus size increases, the type of employed similarity measure plays an important role in determining the changes in the performances. When w_a weight are calculated using the cosine similarity,

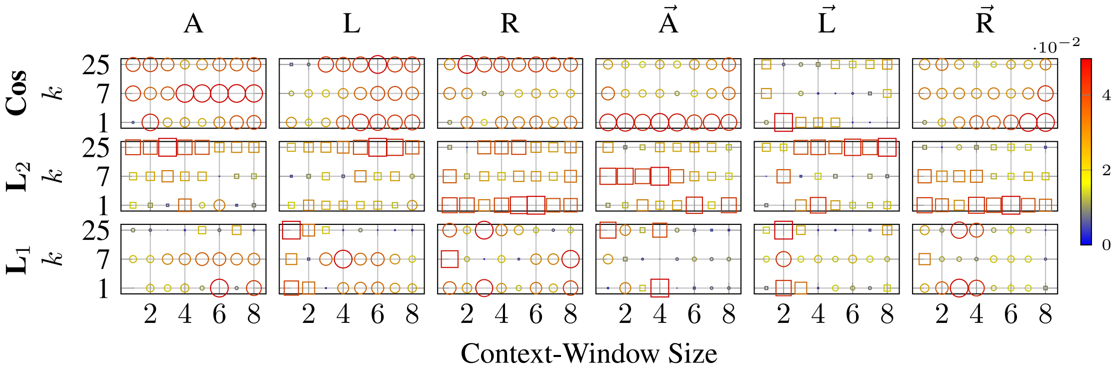
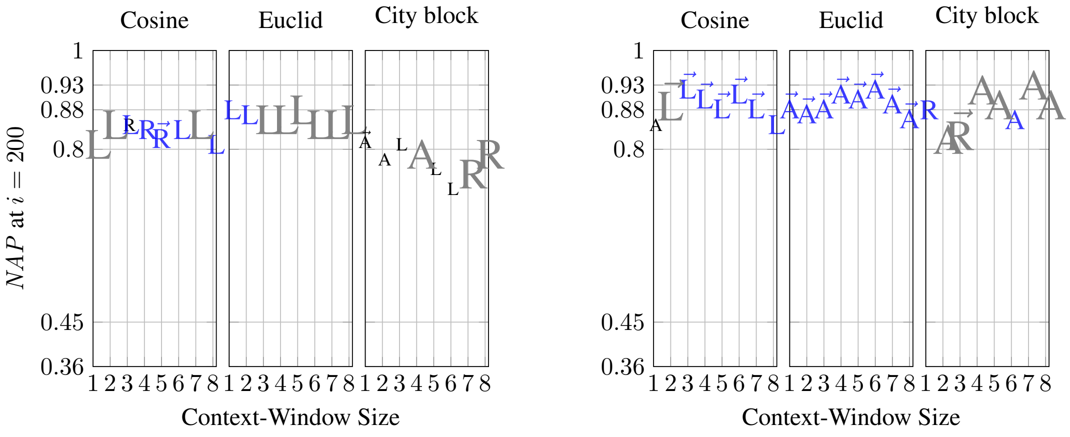


Figure 7: Changes in the performance of models caused by increasing the size of the input corps: the absolute value of the difference between the performance of a model constructed in G_e and G_o are shown. Squares denote negative impacts, while circles show improvements. The size/colour of shapes represents the amount of changes. The x-axis shows various configurations of context-windows (i.e., size, direction, and encoding word order information). The y-axis, however, represents classification parameters (i.e., the values of k and the employed measures for calculating similarities). For instance, when using the cosine similarity for classification in models constructed using context-windows that extend to the Left side of terms, size $= 6$ and $k = 25$, the performance in G_e is 0.448; the same parameters and configuration in G_o gives the performance of 0.40. This increase in the performance is shown by a wide circle in the plot.

enlarging the size of the corpus enhances the performance. Similarly, the city block distance shows a relatively better performance with larger input corpus. However, when similarities are measured using the Euclidean distance, an increase in the size of the corpus can drastically decline the performance. Using additional text, therefore, *does not guarantee* an improvement in the performance.



(a) Obtained performances from G_o

(b) Obtained performances from G_e

Figure 8: The observed performances when using NAP_i at $i = 200$ (i.e., approximately 2% recall). A notation similar to Figure 6 represents the results. In this plot, the minimum value of y-axis, i.e., 0.36, is the computed $NAP_{i=200}$ from the set of c-value ranked candidate terms (i.e., the baseline).

Figures 6 and 7 examine the method's performance for a large recall value. However, in a number of applications, we may be interested only in a small number of terms at the top of these ranked set of terms. Figures 8 and 9, similar to Figures 6 and 7, show the method's performance, however, when it is measured using NAP at $i = 200$ (i.e., for a small recall). In this case, increasing the size of the corpus can enhance or diminish the performance by 20%. Again, compared to the cosine and the city block distance, the Euclidean distance is more susceptible to changes in the corpus size. Specifically, for $k = 1$, the performance frequently drops when the corpus is enlarged.

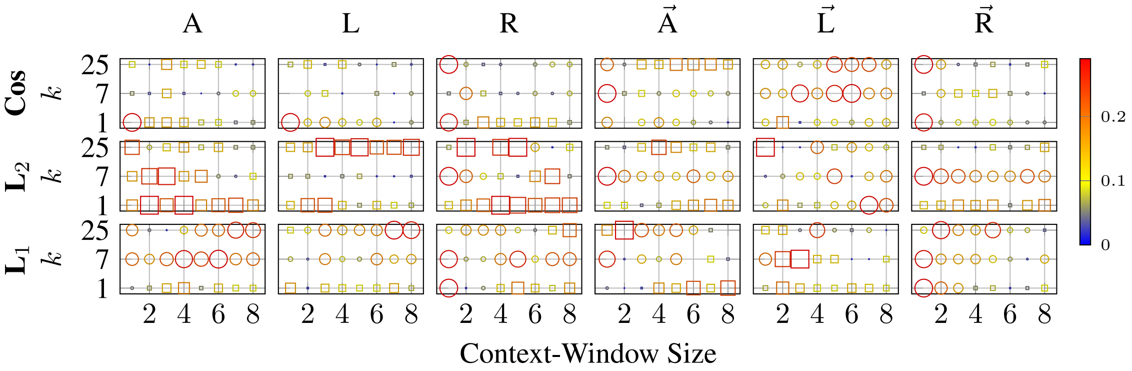


Figure 9: Differences in the performance based on the computed NAP at $i = 200$. The notation is similar to Figure 7.

5 Discussion

We investigated the use of a distributional method for finding co-hyponym terms using a memory-based classification technique. The method is useful when sufficient training data for developing an entity tagger is not available, e.g., when building a terminological resource with a taxonomic structure from scratch. Stable sparse random projections are employed to construct vector spaces directly at a reduced dimensionality. The models are then evaluated for term classification using a k-nn regression framework. We investigated the interplay between the size of the corpus that is used for the construction of the models, the configuration of context-windows (i.e., the way co-occurrence frequencies are collected), and metrics that are employed to measure similarity between vectors.

Our experiments showed that increasing the size of the input corpus for collecting co-occurrence frequencies can improve the performance of the proposed method if suitable configurations of context-windows and similarity metrics are used. We witnessed that the top performer parameters in the original corpus of a small size were not necessarily the top performers when the corpus size increases. In addition, we noticed that choosing the best performing parameters largely depends on the criteria set for the performance assessment. For instance, the city block distance showed a poor performance when the method is assessed at the 100% recall. However, at a small recall point, the city block showed a better performance than other metrics. These observations can perhaps justify a number of contradictory reports in the literature on the effect of the corpus size in the performance of distributional models.

On average, compared to the Euclidean and city block distance, cosine showed a better performance and a more positive and stable response to the increases in the size of the input corpus. This result can be expected intuitively, since cosine shows the degree of commonality between the elements of two vectors. Accordingly, we expect that the reported results can be improved further if, instead of normed-based metrics, a correlation coefficient measure is employed for computing similarities between vectors. Last but not least, a number of influential factors in the obtained results (e.g., the role of R_s and its size, the effect of using linguistic information or indirect co-occurrences) remained unexplored. The entries in specialized vocabularies are rare and less frequent than general vocabularies. For example, a handful of terms in the GENIA corpus (e.g., the term *physiologic cell lineage*) are so rare that they have appeared only once in the abstracts that are pulled out from the PubMed. It is interesting to design an experiment to study the reciprocal between the size of the corpus and the method's performance for the extraction of rare terms. The use of random projection matrix with standard distributions limits the use of common smoothing techniques such as the pointwise mutual information. These can be examined in future work.

Acknowledgments

We would like to thank reviewers for their constructive comments and criticisms. The author of this paper is funded by the Deutsche Forschungsgemeinschaft (DFG) through the 'Collaborative Research Centre 991 (CRC 991): The Structure of Representations in Language, Cognition, and Science'.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 19–27, Stroudsburg, PA, USA. ACL.

Sophia Ananiadou. 1994. A methodology for automatic term recognition. In *Proceedings of the 15th conference on Computational linguistics - Volume 2*, COLING '94, pages 1034–1038, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247. Association for Computational Linguistics.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526.

Scott Cederberg and Dominic Widdows. 2003. Using lsa and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 111–118, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sunandan Chakraborty, Lakshminarayanan Subramanian, and Yaw Nyarko. 2014. Extraction of (key,value) pairs from unstructured ads. In *AAAI Fall Symposium Serie*.

Walter Daelemans and Antal Van Den Bosch, 2010. pages 154–179. Wiley-Blackwell.

Marie Dupuch, Laëtitia Dupuch, Thierry Hamon, and Natalia Grabar. 2014. Exploitation of semantic methods to cluster pharmacovigilance terms. *J. Biomedical Semantics*, 5:18.

Joe Ellis, Xuansong Li, Kira Griffitt, Stephanie M. Strassel, and Jonathan Wright. 2012. Linguistic resources for 2012 knowledge base population evaluations. In *Text Analysis Conference (TAC)*.

KaterinaT. Frantzi, Sophia Ananiadou, and Junichi Tsujii. 1998. The c-value/nc-value method of automatic recognition for multi-word terms. In *Research and Advanced Technology for Digital Libraries*, volume 1513 of *Lecture Notes in Computer Science*, pages 585–604. Springer Berlin Heidelberg.

James Gorman and James R. Curran. 2006. Scaling distributional similarity to large corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 361–368, Stroudsburg, PA, USA. Association for Computational Linguistics.

Benoit Habert, Adeline Nazarenko, Pierre Zweigenbaum, and Jacques Bouaud. 1998. Extending an existing specialized semantic lexicon. In Antonio Rubio, Navidad Gallardo, Rosa Castro, and Antonio Tejada, editors, *Proceedings First International Conference on Language Resources and Evaluation*, pages 663–668, Granada, may.

Kyo Kageura and Bin Umino. 1996. Methods of automatic term recognition: A review. *Terminology*, 3.2 (1996):259–289.

J. . D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, JNLPBA '04, pages 70–75, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marie-Claude L'Homme. 2014. Terminologies and taxonomies. *Oxford Handbooks Online*.

Ping Li. 2007. Very sparse stable random projections for dimension reduction in l_α $(0 < \alpha < 2)$ norm. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 440–449, New York, NY, USA. ACM.

James Mayfield, Paul McNamee, Craig Harmon, Tim Finin, and Dawn Lawrie. 2014. KELVIN: Extracting Knowledge from Large Text Collections. In *AAAI Fall Symposium on Natural Language Access to Big Data*. AAAI Press, November.

George A. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, November.

Hiroshi Nakagawa and Tatsunori Mori. 2002. A simple but powerful automatic term extraction method. In *COLING-02 on COMPUTERM 2002: second international workshop on computational terminology - Volume 14*, COMPUTERM '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chikashi Nobata, Nigel Collier, and Jun ichi Tsujii. 1999. Automatic term identification and classification in biology texts. In *In Proc. of the 5th NLPRS*, pages 369–374.

Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 938–947, Stroudsburg, PA, USA. Association for Computational Linguistics.

Behrang QasemiZadeh and Siegfried Handschuh, 2015. *Random Indexing Explained with High Probability*, pages 414–423. Springer International Publishing, Cham.

Magnus Sahlgren, Anders Holst, and Pentti Kanerva. 2008. Permutations as a means to encode order in word space. In V. Sloutsky, B. Love, and K. Mcrae, editors, *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pages 1300–1305. Cognitive Science Society, Austin, TX.

Magnus Sahlgren. 2005. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005.*

Patrick Schone and Daniel Jurafsky. 2001. Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Burr Settles. 2005. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192.

John Sinclair. 1996. Preliminary recommendations on corpus typology. Technical Report EAGLES Document EAG-TCWG-CTYP/P, EAGLES.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January.

Behrang Q. Zadeh and Siegfried Handschuh. 2014. Random manhattan integer indexing: Incremental l1 normed vector space construction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1713–1723. Association for Computational Linguistics.