Synonym Extraction of Medical Terms from Clinical Text Using Combinations of Word Space Models

Aron Henriksson[1], Hans Moen[2], Maria Skeppstedt[1], Ann-Marie Eklund[3], Vidas Daudaravičius[4], Martin Hassel[1]

[1]Department of Computer and System Sciences (DSV), Stockholm University, Sweden
[2]IDI, Norwegian University of Science and Technology, Norway
[3]Department of Swedish, University of Gothenburg, Sweden
[4]Vytautas Magnus University, Lithuania

Abstract

In information extraction, it is useful to know if two signifiers have the same or very similar semantic content. Maintaining such information in a controlled vocabulary is, however, costly. Here it is demonstrated how synonyms of medical terms can be extracted automatically from a large corpus of clinical text using distributional semantics. By combining Random Indexing and Random Permutation, different lexical semantic aspects are captured, effectively increasing our ability to identify synonymic relations between terms. 44% of 340 synonym pairs from MeSH are successfully extracted in a list of ten suggestions. The models can also be used to map abbreviations to their full-length forms; simple pattern-based filtering of the suggestions yields substantial improvements.

1 Introduction

The choice of words in a text depends on several factors, such as by and for whom it is produced. In a health care setting, for instance, records may reflect both care provider and patient competence, the medical topic of interest and narrative aspects (Rosenbloom et al., 2011). This entails that one and the same concept is often signified by different forms, both through the use of synonyms and abbreviations.

For information extraction to be fully effective, it is important that these alternative lexical instantiations are associated with their corresponding concept in a medical controlled vocabulary, e.g. in UMLS[1]. To develop such terminological resources manually is time-consuming. There is thus a need for (semi-)automatic methods for vocabulary expansion, especially ones that can adapt to the language used in the clinical reality and to different genres of clinical text, as well as to changes over time.

The aim of this study is to investigate and refine a method that fulfills these requirements, namely to apply models of distributional semantics to a large corpus of clinical text. These models quantify the semantic similarity between words based on co-occurrence information and can therefore be used to find potential synonyms of medical terms, as well as to find abbreviation-word pairs. A number of word spaces, based on Random Indexing and Random Permutation – as well their combination in various ways – are evaluated for their ability to extract related medical terms.

2 Background

2.1 Synonyms

Synonyms are different word forms with closely related meanings. They are typically interchangeable in one context but not in another (Yule, 1996). Perfect synonyms – interchangeable in any context – are practically non-existent (Edmonds and Hirst, 2002). Instead, we often speak of near synonyms, which may differ in emphasis, formality or collocational properties (Inkpen and Hirst, 2006).

In domains such as life science and medicine, abbreviations and acronyms are common (Lu et al.,

[1]**U**nified **M**edical **L**anguage **S**ystem: http://www.nlm.nih.gov/research/umls/

2009). Like synonyms, abbreviations and acronyms are interchangeable with their full-length forms.

If perfect synonyms are rare, how may we measure the degree of synonymy, or semantic similarity? Two means of defining semantic similarity are based on topological proximity and statistical measures (Batet et al., 2011). In the first set of approaches, ontological knowledge is taken into account, e.g. by utilizing the taxonomical structure of a biomedical ontology to obtain a measure of similarity between terms, based on the number of shared and non-shared hypernyms. Statistical measures, on the other hand, are typically based on co-occurrence or information content, i.e. the amount of information provided by a given term based on its probability of occurring in a corpus. Semantic similarity is a specialization of the notion of semantic relatedness; Zhang et al. (2012) provide an in-depth survey of related terminology, methods and their evaluation.

2.2 Word Space Models

In information retrieval, the vector space model (Salton et al., 1975) has been successfully applied to tasks such as automatically detecting semantic relations between documents in a collection. However, a fundamental deficiency of this model is that it does not take into account the variability of word choice due to, for instance, synonymy. *Latent Semantic Indexing* (LSI) (Deerwester et al., 1990) is an implementation of the vector space model, created to overcome the problem of synonymy affecting recall by making it possible to detect semantic relations between terms. Hence, it is often appropriately referred to as the word space model. The underlying idea is the *distributional hypothesis* (Harris, 1954), which states that words with similar distribution in language have similar meanings. This allows us to quantify the semantic similarity of words by comparing their distributional profiles. LSI in its original conception has received some criticism for its poor scalability properties, as well as for the fact that the model needs to be completely rebuilt each time new content is added.

Random Indexing (RI) (Kanerva et al., 2000) is a more recent method with better scalability that incrementally builds a word space of contextual information by utilizing co-occurrence information. Each unique term in the corpus is assigned a static *index vector*, consisting of zeros and a small number of randomly placed 1s and -1s. Each term is also assigned an initially empty (only zeros) *context vector*, which is incrementally updated by adding the index vectors of the surrounding words within a sliding window, weighted by their distance to the target term. The size of the window and the dimensionality of the vectors are predefined and remain unchanged throughout indexing.

Random Permutation (RP) (Sahlgren et al., 2008) is a variation of RI that incorporates the same desirable properties as RI, but attempts also to capture term-order information. RP is inspired by the work of Jones and Mewhort (2007) and their *BEAGLE* model that uses vector convolution to incorporate term-order information in the word space model. RP is a computationally lighter alternative to this approach. *Order vectors* are here generated in a similar way as context vectors in RI, but the index vectors of the terms in the sliding window are permuted (i.e. shifted) according to their direction and distance from the target term before they are added to its context vector.

2.3 Related Research

Identification, disambiguation and generation of abbreviations and synonyms have been subjects of interest in several different areas of life-science and medicine (see e.g. Schwartz and Hearst, 2003; Ruiz-Casado et al., 2005; Limsopatham et al., 2011). For instance, a rule-based approach is studied by Conway and Chapman (2012), where variations of synonyms from lexical databases are generated (using term re-ordering, abbreviation generation, etc.) and verified against a corpus of clinical text. This method, however, fails to capture lexical instantiations of concepts that do not share morphemes or letters with the seed term.

Attempts have also been made to utilize structural elements of the documents from which the semantic relations between terms are derived. For instance, Bøhn and Nørvåg (2010) use the internal links in Wikipedia to identify synonyms of named entities. While an effective approach, it is not feasible with health record data since this is not extensively linked.

A viable method for extracting semantic relations from unstructured text is based on the word space model (see Section 2.2), which has performed well on tasks such as taking the TOEFL[2] test. By including order information, RP is shown to outperform RI at this particular task (Sahlgren et al., 2008). This approach of combining RI and RP for extraction of medical synonyms from clinical text has, however, not been studied previously.

3 Method

3.1 General Approach

The general idea behind the approach in this study is to combine multiple word spaces, in which the semantic relations between words have been modeled slightly differently, in an attempt to increase the likelihood of being able to identify synonym pairs. The word spaces are in this case induced from a large corpus of unstructured clinical text.

Two sets of models are built: one based on RI and one based on RP. In addition to using the models separately, we combine them in various ways. In doing so we hope to exploit different semantic properties of words[3] and ultimately boosting the results. We build a number of word spaces of each type with different model parameters. In particular, we experiment with the size of the sliding window, which affects the types of semantic relations that are modeled. We also build RP models with and without stop words[4]. Combining models not only allows us to exploit the advantages of RI and RP, but also to combine models with different parameter settings.

The models and their combinations are then used to generate a list of candidates for synonyms, abbreviations or abbreviation expansions for a given query term. This is carried out by calculating the distributional similarity between the query term and all other terms in the index – i.e. by taking the cosine of the angles between their vectorial representations – and returning the terms closest in the word space. Subsequently, post-processing filtering of the candidates is performed in an attempt to improve the results. The models are evaluated for their ability to detect three types of relations: synonym pairs (**Syn**), abbreviation-expansion pairs (**Abbr→Exp**) and expansion-abbreviation pairs (**Exp→Abbr**).

3.2 Experimental Setup

3.2.1 Data

The data used in these experiments was extracted from the Stockholm EPR Corpus (Dalianis et al., 2009), which contains health records written in Swedish[5]. The data set, from which the models are induced, comprises documents that each contains clinical notes documenting a single patient visit at a particular clinical unit. The data was pre-processed by removing punctuation marks and digits, while lemmatization was done using the *Granska Tagger* (Knutsson et al., 2003). Two versions of the data set were created: one in which stop words have been removed (∼22.5m tokens) and one in which they have been retained (∼42.5m tokens).

3.2.2 Model Combinations

The experiments were aimed at evaluating RI and RP models[6] – built with different parameter settings – and various combinations of the two. For all models, window sizes of two (1+1), four (2+2) and eight (4+4) surrounding words were used. In addition, an RI model with a window size of twenty (10+10) was experimented with, both in isolation and in combination with RP models with window sizes of two and four. The RP models were created with and without stop words. The five models and model combinations are:

Random Indexing (RI). *Context information*. Retrieves terms that have contextually high similarity to the query term; term order is ignored.

Random Permutation (RP). *Order information*. Given a query term, the RP model retrieves terms that have statistically similar neighbors at the same relative positions. This means that the most

[2] Test of English as a Foreign Language

[3] Sahlgren et al. (2008) showed that word spaces containing context vectors and order vectors have only 33% overlap.

[4] As function words are important to the syntactic structure, Jones and Mewhort (2007) include stop words when building models of order information, but not for context information.

[5] This research has been approved by the Regional Ethical Review Board in Stockholm (Etikprövningsnämnden i Stockholm), permission number 2009/1742-31/5.

[6] JavaSDM (http://www.nada.kth.se/∼xmartin/java/) served as the basis for the implementation. We used a dimensionality of 1,000 with 8 non-zero elements in the index vectors. For RI, the weights of the index vectors were calculated as $weight_i = 2^{1-dist_{it}}$, where $dist_{it}$ is the distance to the target word.

similar terms are likely to have similar grammatical properties as the query.

Random Indexing with Random Permutation Filtering (RI_RP). Finds the top ten terms in the RI model that are among the top thirty terms in the RP model. The intuition behind this is to see if synonyms can be detected by trying to make sure that the set of selectable contextually related terms also have similar grammatical properties.

Random Permutation with Random Indexing Filtering (RP_RI). Finds the top ten terms in the RP model that are among the top thirty terms in the RI model. This is included to see if using the opposite of the above model order will yield any interesting results.

Random Indexing + Random Permutation, (RI+RP). Sums the similarity scores of each suggestion for the query as given by the two models.

3.2.3 Post-Processing

In order to discard poor suggestions automatically and retrieve potentially better ones, the following post-processing, or filtering, rules were constructed:

$$Syn = \begin{cases} True & \textit{if } (Cos \geq 0.60) \vee (Cos \geq 0.40 \wedge Rank < 9) \\ False & Otherwise \end{cases}$$

$$Exp \rightarrow Abbr = \begin{cases} True & \textit{if } (Len < 5) \wedge (Sub_{out} = True) \\ False & Otherwise \end{cases}$$

$$Abbr \rightarrow Exp = \begin{cases} True & \textit{if } (Len > 4) \wedge (Sub_{in} = True) \\ False & Otherwise \end{cases}$$

Cos : *Cosine similarity between suggestion and query term.*
$Rank$: *The rank of the suggestion, ordered by cosine similarity.*
Sub_{out} : *Whether each letter in the suggestion is present*
in the query term, in the same order, and the initial letter identical.
Sub_{in} : *Whether each letter in the query term is present.*
in the suggestion, in the same order, and the initial letter identical.
Len : *The length of the suggestion.*

For synonym extraction, rank and cosine similarity thresholds were set to maximize precision, without negatively affecting recall. For valid abbreviations/expansions, each letter in the abbreviation had to be present in the expanded word (in the same order), while the length of abbreviations and expansions was restricted. Evaluated on the development set for the their ability to classify the top-ten suggestions as correct/incorrect, the post-processing rules obtained the following results: *Syn*: 0.051 precision, 1 recall; *Abbr→Exp*: 0.34 precision, 0.98 recall; *Exp→Abbr*: 0.89 precision, 1 recall.

The post-processing filtering rules were employed in two different ways. In the first approach, the models were forced to make a predefined number (ten) of suggestions, irrespective of how good they were deemed to be by the model. Suggestions were retrieved by the model until ten had been classified as correct according to the post-processing rules or one hundred suggestions had been processed. If less than ten were classified as incorrect, the highest ranked discarded terms were used to populate the remaining slots in the final list of suggestions. In the second approach, the models were allowed to suggest a dynamic number of terms, with a minimum of one and a maximum of ten. If none of the highest ranked terms were classified as correct, the highest ranked term was suggested.

3.2.4 Evaluation

Known abbreviation-expansion pairs and known synonyms were used as test data, and the models were evaluated for their ability to produce the expected abbreviation/expansion/synonym among ten suggestions. Test data for abbreviations was derived from Cederblom (2005), while the Swedish version of MeSH[7] and its extension (KIB, 2012) were used for synonyms (Table 1).

Since neither the meaning of multiword expressions nor very rare words can be captured by the constructed models, only pairs of unigrams that occurred at least fifty times in the corpus were used as test data. Moreover, hypernym/hyponym and other non-synonym pairs found in the UMLS version of MeSH were also removed from the test data.

The models were tested using each abbreviation, abbreviation expansion and synonym in the development set as a query; recall for including the corresponding abbreviation(s)/expansion(s) or synonym(s) in the top ten suggestions was measured.

[7]**Me**dical **S**ubject **H**eadings (`http://www.ncbi.nlm.nih.gov/mesh`) is a part of UMLS.

Set Type	Size	2 cor	3 cor
Abbr→Exp (Development)	117	9.4%	0.0%
Abbr→Exp (Evaluation)	98	3.1%	0.0%
Exp→Abbr (Development)	110	8.2%	1.8%
Exp→Abbr (Evaluation)	98	7.1%	0.0%
Syn (Development)	334	9.0%	1.2%
Syn (Evaluation)	340	14%	2.4%

Table 1: Test data was randomly split into a *development set* and an *evaluation set*. *Size* shows the number of queries, *2 cor* shows the proportion of queries with two correct answers and *3 cor* the proportion of queries with three correct answers. The remaining queries have one correct answer.

For the model and parameter setting that yielded the best results for each of the three query types, different post-processing techniques were evaluated on the development data. Finally, recall and precision for the best models were measured on the evaluation data, both before and after post-processing.

4 Results

The optimal model parameters vary across models and model combinations, as can be seen in Table 2. It also depends on which task the models are applied to. The best results for all three tasks are obtained when the similarity scores of RI and RP are combined (RI+RP). For abbreviation expansion (abbr→exp), the best result is obtained when a sliding window of size four is used in both models, with the RP model trained on the data set that includes stop words: 0.42 recall. This model configuration is also the most successful when matching full-length words to their abbreviated forms (exp→abbr), although a larger window size of eight yields the same result: 0.32 recall. For identifying synonyms (syn), however, combining an RI model with a significantly larger window size (twenty) and an RP model with stop words and a window size of four yields the best result: 0.40 recall.

In an attempt to identify general trends for individual parameter settings, average scores have been calculated where a single variable is held constant at a time. In Table 3, the various models and their combinations are evaluated without considering any of the model parameters. As expected, the RI+RP combination is the most successful overall.

In Table 4, the effect of the window size is evaluated. Here there is no clear tendency, except that a window size of four or eight seems to work relatively well. However, combining an RI model with a very large window size (twenty) and an RP model with a window size of four works equally well.

In Table 5, the effect of applying RP models on data with or without removing stop words is evaluated. Although the best model combinations almost invariably include the use of an RP model with stop words, when looking at the average across all models and their combinations, there is little difference.

Based on the results in Table 2 – evaluated on the development set – the best model configurations were selected. This data set was also used to generate the post-processing rules. To evaluate the generalizability of the model selection and the post-processing, they were applied to the evaluation set (Table 6). Compared to the preliminary results, the performance of the models dropped quite a bit on the abbr→exp and exp→abbr tasks, with 0.31 and 0.20 recall respectively. On the synonym task, however, recall increased from 0.40 to 0.44.

Model	Abbr→Exp	Exp→Abbr	Syn
RI	0.32	0.25	0.36
RP	0.33	0.27	0.31
RP_RI	0.33	0.28	0.35
RI_RP	0.34	0.27	0.33
RI+RP	0.37	0.29	0.37

Table 3: Average results (recall, top ten) of the models and their combinations on the three tasks.

Sl. Window	Abbr→Exp	Exp→Abbr	Syn
2	0.31	0.23	0.30
4	0.35	0.29	0.35
8	0.36	0.29	0.37
20 RI, 2 RP	0.34	0.27	0.34
20 RI, 4 RP	0.36	0.29	0.37

Table 4: Average results (recall, top ten) of the models with different (sliding) window sizes on the three tasks.

The post-processing yields a substantial improvement on recall for abbreviations: 11 (abbr→exp) and 13 (exp→abbr) percentage points respectively on the two tasks. For synonyms, however, there is

Model	Abbr→Exp		Exp→Abbr		Syn	
	Best Config.	Result	Best Config.	Result	Best Config.	Result
RI	RI_4	0.33	RI_4/8	0.26	RI_20	0.39
RP	RP_4/8	0.37	RP_8	0.31	RP_4/8	0.35
RP_RI	RI_20, RP_4_sw	0.35	RI_4/20, RP_4_sw	0.30	RI_8, RP_8_(sw)	0.38
					RI_20, RP_2_sw	0.38
RI_RP	RI_8, RP_8_sw	0.38	RI_8, RP_8	0.30	RI_8, RP_8	0.39
RI+RP	RI_4, RP_4_sw	**0.42**	RI_4, RP_4_sw	**0.32**	RI_20, RP_4_sw	**0.40**
			RI_8, RP_8_sw	**0.32**		

Table 2: Results (recall, top ten) of the best configurations for each model (combination) on the three tasks. The configurations are described according to the following pattern: *model_windowSize*. For RP, *sw* means that stop words are retained in the model. A slash means that either configuration could be used; brackets indicate an optional configuration.

Stop Words	Abbr→Exp	Exp→Abbr	Syn
RP w/ SWs	0.35	0.28	0.34
RP w/o SWs	0.34	0.27	0.35

Table 5: Average results (recall, top ten) of the RP models (and their inclusion in the model combinations) with or without stop words on the three tasks.

Frequency Threshold	Recall (95% CI)
50	0.40 (± 0.05)
100	0.46 (± 0.07)
200	0.49 (± 0.09)
300	0.53 (± 0.10)
400	0.54 (± 0.12)
500	0.52 (± 0.13)

Table 7: Recall values with lower thresholds for the number of occurrences of the synonym pairs in the data.

no improvement; in fact, the recall drops a percentage point. When allowing the models to suggest a dynamic number of terms – but at most ten – it is not possible to improve on the recall obtained by the previous post-processing option. Instead, improved precision is the aim of this mode. For abbreviations, precision increases by three (abbr→exp) and seven (exp→abbr) percentage points respectively. Again, no improvement is observed for synonyms. It should be noted that this option may have a negative impact on recall; however, since recall was maximized at the expense of precision when designing the rules, the impact is in this case almost negligible.

To investigate how term frequency may affect performance, results on the synonym tasks are reported based on frequency thresholds for the synonym pairs (Table 7). This shows that results improve as the number of observations in the data increases, up until the five hundred frequency mark.

5 Discussion

For both synonym extraction and abbreviation-word mapping, combining RI and RP yields improvements over using only one of the models in isolation, indicating that contextual and more order-dependent relations supplement each other in such tasks. Thus, if two lexical items are both distributionally similar and share grammatical properties, they are more likely to be synonymous.

Although another advantage of this approach is enabling models with different parameter settings to be combined, the best results on the two abbreviation tasks were obtained by combining RI and RP models with identical window sizes (four or eight). The optimal window sizes in these experiments are roughly in line with those reported by Sahlgren et al. (2008), suggesting that synonym extraction by means of distributional semantics may be transferable across domains. On the synonym task, combining a large-context RI model with a somewhat smaller window in the RP model performed best; however, the improvement yielded by this combination is almost negligible. The only conclusion that can be made with some confidence is that a window size of two is too small when performing these

	Abbr→Exp		Exp→Abbr		Syn	
	P	R	P	R	P	R
Model	RI_4+RP_4_sw		RI_4+RP_4_sw		RI_20+RP_4_sw	
Without post-processing	0.05	0.31	0.03	0.20	0.07	0.44
With post-processing	0.08	0.42	0.05	0.33	0.08	0.43
Dynamic # of suggestions	0.11	0.41	0.12	0.33	0.08	0.42

Table 6: Results (P = precision, R = recall, top ten) of the best models with and without post-processing on the three tasks. Dynamic # of suggestions allows the model to suggest less than ten terms in order to improve precision. The results are based on the application of the model combinations to the evaluation data.

tasks. By the same token it is hard to state something definite regarding the choice of whether to include stop words in the RP models. However, since most of the best model combinations include an RP model with stop words, they seem to contribute to capturing the grammatical properties of neighboring words and are useful given that sufficient (semantically valuable) contextual information – provided by the RI model without stop words – is available.

Since the targeted application was primarily semiautomatic development of terminologies – in which useful additions are manually selected from a reasonable number of candidates – a system able to suggest correct synonyms in almost half of the cases is useful. It should be noted, however, that many of the suggestions that were labeled as incorrect were nevertheless reasonable candidates, e.g. spelling variants, which would be desirable to identify from an information extraction perspective. Many of the suggestions were, however, non-synonymous terms belonging to the same semantic class as the query term, such as drugs, family relations, occupations and diseases. In general, the more frequently occurring synonym pairs were easier to detect. The ones that were not detected could perhaps be explained by homonymity and synonym pairs being preferred by different medical professions – these may not always occur in the same context.

Better results are achieved for synonym extraction than for abbreviation expansion, although the latter is intuitively a less complex task. This is probably due to the ambiguity of medical abbreviations, entailing that abbreviations in the clinical corpus often have a meaning other than the expansion included in the test data. The simple post-processing filtering of candidate terms was, however, as expected, more effective for abbreviations than synonyms. An improvement of over ten percentage points in this case is substantial and demonstrates the potential of such techniques for abbreviation expansion.

Directions for future work could include exploring additional variants of RI, e.g. *direction vectors* (Sahlgren et al., 2008) and *Reflective Random Indexing* (Cohen et al., 2010), as well as improving the post-processing filtering. A limitation of the current models is that they are restricted to unigrams; this needs to be addressed, as many synonym pairs are multiword expressions of varying length.

6 Conclusion

Our experiments show that multiple word space models can be combined to improve automatic extraction of synonym candidates and abbreviation-word pairs from clinical text. The best results are achieved by summing the similarity scores from Random Indexing and Random Permutation models. Further improvements are made in the abbreviation-word mapping task by applying a set of simple post-processing rules. Although there is room for improvement, this study demonstrates that this approach can serve as useful terminology development support in the medical domain.

Acknowledgements

We would like to thank the funders – Nordforsk and the Nordic Council of Ministers – and members, especially Dr. Maria Kvist, of our research network HEXAnord. We are also grateful to Staffan Cederblom and Studentlitteratur for giving us access to their database of medical abbreviations. We would also like to thank the reviewers for their valuable comments.

References

Montserrat Batet, David Sánchez and Aida Valls. 2011. An ontology-based measure to compute semantic similarity in biomedicine. Journal of Biomedical Informatics, 44(1): 118–125.

Christian Bøhn and Kjertil Nørvåg. 2009. Extracting Named Entities and Synonyms from Wikipedia. In Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications, AINA 2010, Perth, Australia, 20-13 April 2010, pp. 1300–1307.

Staffan Cederblom. 2005. Medicinska förkortningar och akronymer. Studentlitteratur, Lund, Sweden.

Trevor Cohen, Roger Schvaneveldt and Dominic Widdows. 2010. Reflective Random Indexing and indirect inference: a scalable method for discovery of implicit connections. Journal of Biomedical Informatics, 43(2): 240–56.

Mike Conway and Wendy Chapman. 2012. Discovering Lexical Instantiations of Clinical Concepts using Web Services, WordNet and Corpus Resources. AMIA Fall Symposium 2012.

Hercules Dalianis, Martin Hassel and Sumithra Velupillai. 2009. The Stockholm EPR Corpus: Characteristics and Some Initial Findings. In Proceedings of ISHIMR 2009, pp. 243–249.

Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer and Richard Harshman. 1990. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6): 391–407.

Philip Edmonds and Graeme Hirst. 2002. Near-synonymy and lexical choice. Computational Linguistics, 28(2): 105–144.

Zellig S. Harris. 1954. Distributional structure. Word, 10: 146–162.

Diana Inkpen and Graeme Hirst. 2006. Building and Using a Lexical Knowledge Base of Near-Synonym Differences. Computational Linguistics, 32(2): 223–262.

Michael N. Jones and Douglas J. K. Mewhort. 2007. Representing word meaning and order information in a composite holographic lexicon. Psychological Review, 114(1): 1–37.

Pentti Kanerva, Jan Kristofersson and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. In Proceedings of the 22nd Annual Conference of the Cognitive Science Society, pp. 1036.

KIB, Karolinska Institutet University Library 2012, Svensk MeSH - MeSH sökverktyg, accessed 14 May 2012, <http://http://mesh.kib.ki.se/swemesh/swemesh_se.cfm>.

Ola Knutsson, Johnny Bigert and Viggo Kann. 2003. A Robust Shallow Parser for Swedish. In Proceedings of Nodalida 2003.

Nut Limsopatham, Rodrygo L. T. Santos, Craig Macdonald and Iadh Ounis. 2011. Disambiguating biomedical acronyms using EMIM. Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pp 1213-1214.

Yue Lu, Hui Fang and Chengxiang Zhai. 2009. An empirical study of gene synonym query expansion in biomedical information retrieval. Information Retrieval, 12(1): 51–68.

Trent S. Rosenbloom, Joshua C. Denny, Hua Xu, Nancy Lorenzi, William W. Stead and Kevin B. Johnson. 2011. Data from clinical notes: a perspective on the tension between structure and flexible documentation. Journal of the American Medical Informatics Association, 18(2): 181–186.

Maria Ruiz-Casado, Enrique Alfonseca and Pablo Castells. 2005. Using context-window overlapping in synonym discovery and ontology extension. Proceedings of RANLP 2005.

Magnus Sahlgren. 2006. The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. In PhD thesis Stockholm University, Stockholm, Sweden.

Magnus Sahlgren, Anders Holst and Pentti Kanerva. 2008. Permutations as a Means to Encode Order in Word Space. In Proceedings of The 30th Annual Meeting of the Cognitive Science Society, pp. 1300–1305.

Gerald Salton, Andrew Wong and Chung-Shu Yang. 1975. A Vector Space Model for Automatic Indexing. Communications of the ACM, 18(11): 613620.

Ariel S. Schwartz and Marti A. Hearst. 2003. A Simple Algorithm For Identifying Abbreviation Definitions in Biomedical Text. Pacific Symposium on Biocomputing 2003, pp 451-462.

George Yule. 1996. The Study of Language. Cambridge University Press, Cambridge, UK.

Ziqi Zhang, Anna Lisa Gentile and Fabio Ciravegna. 2012. Recent advances in methods of lexical semantic relatedness – a survey. Natural Language Engineering, Available on CJO doi:10.1017/S1351324912000125.