The company that words keep…

Ágoston Tóth*

Institute of English and American Studies, University of Debrecen

Abstract

The extraction of large-scale lexical co-occurrence statistics from huge corpora makes it possible to explore the syntagmatic and paradigmatic relationships between words. Statistics of this type can be computed automatically without human supervision. This technique has been used in the literature to solve certain semantic tasks. This paper reports on an experiment in which context vectors storing co-occurrence data were built for 30 target words, then the system chose the most similar word for each of the target words automatically. After training the model, semantically related words, including synonyms, antonyms and hypernyms/hyponyms were found by the system automatically with 90% precision.

Keywords: distributed semantics, lexical semantics, computational linguistics

1. Introduction

"You shall know a word by the company it keeps" – this often-quoted statement from Firth (1957) is also a motto for many linguists who think that the syntagmatic and paradigmatic context that a word has may have a fundamental and decisive role in defining its meaning. Word co-occurrence statistics can be collected by humans (during language acquisition) and also by computers, and these data can serve as the basis for solving certain semantic tasks (Lund and Burgess, 1996; McDonald and Shillcock, 2001; Turney and Pantel, 2010).

When we use words, we put them into the context of other words; this context establishes immediate *syntagmatic relations* among words. In sentence 1, we observe an instance of a syntagmatic link between the words *drink* and *coffee*.

1) I drink coffee.
2) He drinks milk.
3) Mary sips tea.

* Tóth Ágoston is a Lecturer at the Institute of English and American Studies of the University of Debrecen, deputy head of the Department of English Linguistics. His PhD dissertation ("Perspectives on the lexicon," Budapest, Akadémiai Kiadó, 2008) is a multidisciplinary study that tackles issues in generative syntax, lexical semantics, large-scale lexical databases (WordNet, FrameNet) and contains original research in connectionist machine learning (an artificial neural network that acquires FrameNet frame and frame element labels). His recent publications focus on lexical ambiguity, treebank design and distributed semantics. Email: toth.agoston@arts.unideb.hu

The study of syntagmatic relations is directly useful in finding lexical and phraseological patterns, multi-word expressions (e.g. compounds, light verbs), collocations and idioms.

In sentences 1-3, the words in a single vertical column (e.g. *coffee, milk* and *tea*) form an associative class. These classes are held together by *paradigmatic relations* which are based on the observation that they can co-occur with the same words (e.g. *tea* can co-occur with *drink* and *sip; coffee* can co-occur with these words, too). The widely accepted explanation of this phenomenon is statistical in nature: these words frequently co-occur with the same words (and this is how humans can acquire these classes, too). Please note that the words *I, he* and *Mary,* as well as the words *drink* and *sip,* also form paradigmatic classes. Finally, notice that words in the same paradigmatic class do not typically occur in each other's context, i.e. they do not typically develop syntagmatic (first-order co-occurrence) relations.

In the practice of Computational Linguistics, lexical meaning can be encoded by hand or can be extracted from texts via machine learning. *WordNet* is a lexical database that contains lexical semantic information (e.g. synonymy, antonymy, hypernymy, hyponymy) encoded by hand. A different database, *FrameNet*, contains lexical units associated with frames (e.g. buying, selling) and frame elements (seller, goods, money, etc.) These databases are great resources, but they are very expensive to compile and to maintain while it is a very long process, too.

Extracting semantic information from text corpora via machine learning is cost and time effective. A computer program extracts statistical information from word co-occurrence patterns found in a large text corpus in a way that the similarity of words can be measured. Knowledge extracted via machine learning is certainly much less controlled (and linguistically less transparent and reliable) than a manually collected database, but it may be the only solution if relevant hand-made databases are not available. Also note that machine learning may even help us discover patterns not detected by the human eye.

2. How to build semantic representations automatically?

We develop contextual representations for selected *target* words and we also have to choose words that the system locates in the context of the target words (we will refer to these words as *context words*). In most cases, we work with a few target words (typically 10-

100) and a much larger number of context words. In theory, we can use all words that occur in the context of the target words, but it slows down processing considerably, and the performance gain is minimal (if there is any).

During the process, we build a multi-dimensional vector for each target word, and each context word has its own position in the context vector (thereby creating its own dimension). The value in a given vector position reflects how many times the given context word co-occurred with that target word. E.g. if the word *tea* is a target word, and the word *the* is a context word, and *the* occurs 100 times in the close vicinity (within a "context window") of *tea*, then the vector element corresponding to the word *the* will be set to 100 in the vector of *tea.*

We usually collect the vectors into a matrix, where each row is a context vector for a single target word.

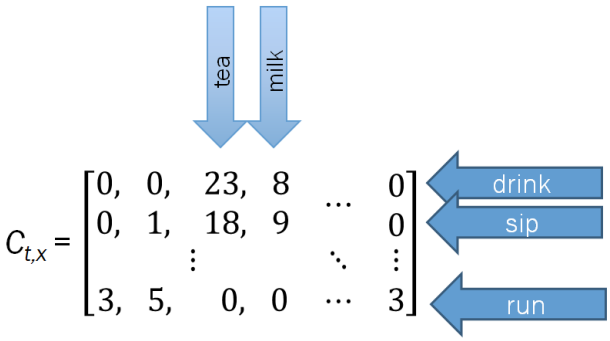


Figure 1 The context matrix

What does "the context of the target word" exactly mean? We tend to use a symmetrical moving window centered on every occurrence of the target word in the text and it contains a few words form its direct left and right context. We can work with the two direct neighbours of the target word (1+1 word context window) or we can consider more words. If a context word appears in this window then we modify the context matrix.

A large corpus (30 million words or more) seems to be necessary for this type of investigation, but the corpus can be unprocessed, "raw" (without lemmatization, part of speech information, etc.).

3. How can we use vector representations?

Having developed the context matrix, we can now compare the context vectors representing the target words. We do this under the presupposition that a similar context is a sign of

similarity in meaning. The context vectors themselves represent syntagmatic links while the comparison phase discovers the paradigmatic relations.

There are two basic methods for the comparison of the context vectors: we can measure vector distances (Figure 2) or the cosine of the angle between vectors (Figure 3). The latter has the advantage that we can compensate for vector length differences, which is useful, since length depends on the frequency of context words and, because of this, it also depends on the frequency of the target word itself, which is a problem if you try to detect a paradigmatic relation between a frequent and a rare word. In my experiment, I used the cosine similarity measure.

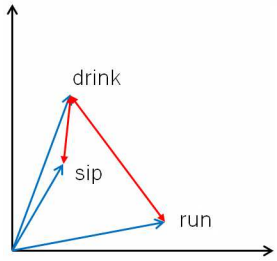


Figure 2 Vector similarity: distance

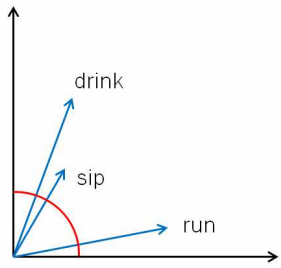


Figure 3 Vector similarity: cosine

Please note that the context vectors are multi-dimensional objects (basically, there are as many dimensions as context words); Figures 2 and 3 are two-dimensional illustrations only.

Armed with the ability of calculating vector similarity, we can now solve a task. A typical task is to find the most similar target word to a selected word (this is the approach I chose in my experiment, which will be described in the next chapter). Carrying out other tasks is also possible, e.g. clustering or visualizing the similarity of words by creating a "word cloud".

4. My vector space experiment

The experiment described below was organized along the lines presented in sections 2 and 3. The necessary software infrastructure was also developed by me.

As far as the training corpus is concerned, I used an 80 million word subcorpus of the Hungarian Webcorpus (Kornai et al., 2006). By repeatedly measuring the precision of the system during the experiment, I could follow the effect of the corpus size up to a point where precision stopped improving. The corpus contained no annotations and no

lemmatization. I kept the context window very small, just one word on the left and one word on the right of the target word. When a context word was found in this small window around the actual target word, the context vector of that target word was modified accordingly.

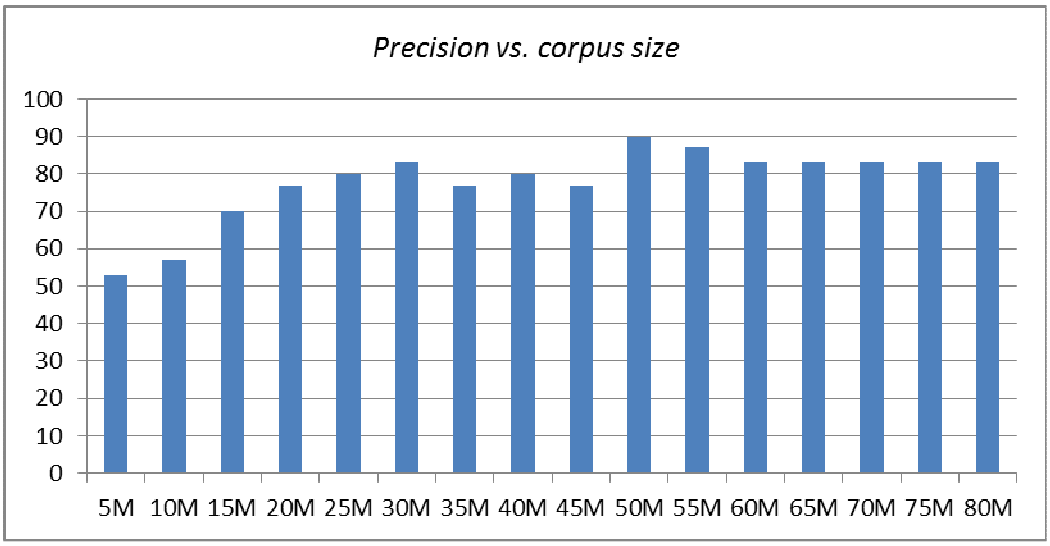
I chose 30 target words: 5 synonym pairs, 5 antonym pairs and 5 pairs that contained hypernyms/hyponyms (superordinate/subordinate terms). These relations and the very fact that words formed semantic pairs were not known to the system in the training phase.

The set of context words contained the 12000 most frequent words of the webcorpus. I did not exclude function words. The context matrix had 30 x 12000 elements (30 target word vectors each containing 12000 context word positions). Having processed the corpus, I normalized the vectors by replacing pointwise mutual information (PMI) values for the raw frequency values (Manning and Schütze, 1999) to fight problems that result from high-frequency target words co-occurring with low-frequency context words, or vice versa.

After computing and normalizing the context vectors, the system measured vector similarity to find the most similar word to each target word. The output of this process was checked against the list of expected results (the synonym/antonym/hyponym/hypernym pair of the target word). Precision figures were reported after processing 5, 10, 15, 20, 25, 30, … 80 million words from the corpus.

Since we have 30 target words, but we do not consider the original target word as a potential candidate for being the most similar word, the random chance of picking out the right alternative is 1:29, 3.4%. This is the initial performance of the system, before training.

Having trained the system with 50 million words, I could measure a 90% precision peak. Then the precision fell back slightly to 83%. I noticed that the optimum corpus size seemed to depend on the number of context words we used, but I have not looked into this issue in a systematic way. The following diagram shows the change of the precision value during the training process:



The next table shows all target words, the closest word to each target after training the system with 50 million words, the expected result (the synomym, antonym or hypernym/hyponym pair of the target word) and whether the result was correct.

Target	**Output**	**Expected result**	**Correct?**
fekete *(black)*	fehér *(white)*	fehér	Y
fehér *(white)*	fekete *(black)*	fekete	Y
régi *(old)*	új *(new)*	új	Y
új *(new)*	régi *(old)*	régi	Y
fent *(to be up)*	lent *(to be down)*	lent	Y
lent *(to be down)*	fent *(to be up)*	fent	Y
ki *(out)*	be *(in)*	be	Y
be *(in)*	ki *(out)*	ki	Y
jó *(good)*	rossz *(bad)*	rossz	Y
rossz *(bad)*	jó *(good)*	jó	Y
legmagasabb *(highest)*	legnagyobb *(greatest)*	legnagyobb	Y
legnagyobb *(greatest)*	teljes *(complete)*	legmagasabb *(highest)*	N
egész *(whole/complete)*	teljes *(complete)*	teljes	Y
teljes *(complete)*	egész *(whole/complete)*	egész	Y
tép *(tears/rips)*	szakít *(tears/rips)*	szakít	Y
szakít *(tears/rips)*	tép *(tears/rips)*	tép	Y

néz *(sees)*	figyel *(watches)*	figyel	Y
figyel *(watches)*	néz *(sees)*	néz	Y
fut *(runs)*	rohan *(rushes)*	rohan	Y
rohan *(rushes)*	fut *(runs)*	fut	Y
alma *(apple)*	gyümölcs *(fruit)*	gyümölcs	Y
gyümölcs *(fruit)*	alma *(apple)*	alma	Y
szekrény *(cupboard)*	épület *(building)*	bútor *(furniture)*	N
bútor *(furniture)*	szekrény *(cupboard)*	szekrény	Y
ház *(house)*	épület *(building)*	épület	Y
épület *(building)*	ház *(house)*	ház	Y
tenisz *(tennis)*	alma *(apple)*	sport *(sport/game)*	N
sport *(sport/game)*	tenisz *(tennis)*	tenisz	Y
dollár *(dollar)*	deviza *(foreign exchange)*	deviza	Y
deviza *(foreign exchange)*	dollár *(dollar)*	dollar	Y

The system was able to reach a very high precision level in this experiment. The reason why *tenisz (tennis)* became more associated with *alma (apple)* than it was with *sport* is not known, but a closer examination of the resulting similarity values shows that *sport* also had a relatively high similarity score (it came out as second). The same applies to the words *greatest* and *cupboard*: the expected word had the second largest similarity score. We could, perhaps, treat the similarity scores as probability values and consider the top candidates in each case.

Another finding of the experiment is that synonyms, antonyms and hypernyms/hyponyms work very similarly and equally well in this experiment type.

Also note that the target and the context words were not lemmatized; I worked with actual word forms throughout the experiment. Hungarian has a very rich morphology, yet the algorithm performed well without lemmatization. Nevertheless, rare forms may be problematic to handle without lemmatization due to the data sparseness problem; this issue is yet to be investigated.

A logical next step is to extend the list of potential target words to see if the precision of the system remains high. This is important since in actual applications (e.g. in information

retrieval), we may need to work with hundreds or thousands of words. I would also like to see what happens if multiple linguistically related words are present in the data (two or more near synonyms, hypernyms and hyponyms for the same target word). Finally, the question of lexical ambiguity (how to detect and handle it) will also be an interesting issue to investigate.

Acknowledgements

The author acknowledges that the research reported here is supported, in part, by OTKA (Hungarian Scientific Research Fund), grant number: K 72983 and by the TÁMOP 4.2.1./B-09/1/KONV-2010-0007 project. The project is co-financed by the European Union and the European Social Fund.

Works Cited

Firth, J. R. "A synopsis of linguistic theory, 1930-1955". In J. R. Firth et al. (eds.) *Studies in Linguistic Analysis*. Oxford: Blackwell, 1957.

Kornai, A, Halácsy, P, Nagy, V, Oravecz, Cs, Trón, V, and Varga, D. "Web-based frequency dictionaries for medium density languages". In Kilgarriff, A. and Baroni, M. (eds.) *Proceedings of the 2nd International Workshop on Web as Corpus*, pp. 1-9, 2006.

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

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

McDonald, S. A., Shillcock, R. C. "Rethinking the word frequency effect: The neglected role of distributional information in lexical processing". *Language and Speech* 44, pp. 295-323, 2001.

Turney, P.D., Pantel, P. "From frequency to meaning: Vector space models of semantics." *Journal of Artificial Intelligence Research (JAIR),* 37, pp. 141-188, 2010.