AI-KU: Unsupervised Co-Occurrence Model for Cross-Level Semantic Similarity

Osman Başkaya
Artifical Intelligence Laboratory
Koç University, Istanbul, Turkey
obaskaya@ku.edu.tr

Deniz Yuret
Artifical Intelligence Laboratory
Koç University, Istanbul, Turkey
dyuret@ku.edu.tr

Abstract

In this paper, we describe our unsupervised method submitted to Cross-Level Semantic Similarity task in Semeval 2014 that aims to compute semantic similarity between two different sized text fragments. Our method models each text fragment by using the co-occurrence statistics of either occurred words or their substitutes. The co-occurrence step provides dense, low-dimensional embedding for each fragment and semantic similarity between embeddings can then be calculated various similarity metrics. Although our current model avoids the syntactic information, we achieved promising results and outperformed all baselines.

1 Introduction

Semantic similarity is a measure that specifies the similarity of one text's meaning to another's. Semantic similarity plays an important role in many Natural Language Processing (NLP) tasks ranging from textual entailment (Berant et al., 2012) to summarization (Lin and Hovy, 2003), question answering (Surdeanu et al., 2011) to text classification (Sebastiani, 2002), word sense disambiguation (Schütze, 1998) to information retrieval (Park et al., 2005).

There are three main approaches to compute semantic similarity between two text fragments. First approach uses Vector Space Models (see Turney & Pantel (2010) for an overview) where each text is represented as a bag-of-word vector. The similarity between two text fragments can then be computed with various metrics such as cosine similarity. For these models, sparseness in the input nature is the key problem. Therefore, later works such as LSI (Deerwester et al., 1990) and Topic Models (Blei et al., 2003) overcome sparsity problem reducing the dimensionality of the model by introducing latent variables. Second approach blends various lexical and syntactic features and attacks the problem with machine learning models. The third approach is based on word-to-word similarity alignment (Pilehvar et al., 2013), (Islam and Inkpen, 2008).

ID	Lemma
R2	how
R2	blog
R2	popular

Table 1: Input example for AI-KU$_1$.

Cross-Level Semantic Similarity (CLSS) task in SemEval 2014 (give a *Reference*) provides an evaluation framework for assessing similarity methods for texts in different volumes (i.e., lexical levels). Unlike previous SemEval and *SEM tasks that have interested in comparing texts with similar volume, this task contains four subtasks (paragraph2sentence, sentence2phrase, phrase2word and word2sense) in order to investigate the performances of the systems on pair of texts with different sizes. A system should report similarity score of a given pair, ranging 4 (the two items have very similar meanings and the most important ideas, concepts, or actions in the larger text are represented in the smaller text) to 0 (the two items do not mean the same thing and are not on the same topic).

In this paper, we describe our two unsupervised systems that based on co-occurrence statistics of words. There is only one difference between the systems is that the input they use. First system uses the words directly (after lemmatization, stop-word removal and excluding the non-alphanumeric characters) in text while the second system utilizes the most likely substitutes consulted by a 4-gram language model for each ob-

	System	Pearson	Spearman
Paragraph-2-Sentence	Best	0.837	0.821
	2^{nd} Best	0.834	0.820
	3^{rd} Best	0.826	0.817
	AI-KU$_1$	0.732	0.727
	AI-KU$_2$	0.698	0.700
	LCS	0.527	0.613
	lch	0.125	0.114
	jcn	0.107	0.106
	JI	0.640	0.687

Table 3: Paragraph-2-Sentence subtask scores for the test set. *Best* indicates the best score for the subtasks. LCS stands for Normalized Longest Common Substring. Subscripts in AI-KU systems indicate the run number.

	System	Pearson	Spearman
Sentence-2-Phrase	Best	0.777	0.642
	2^{nd} Best	0.771	0.760
	3^{rd} Best	0.760	0.757
	AI-KU$_1$	0.680	0.646
	AI-KU$_2$	0.617	0.612
	LCS	0.562	0.626
	lch	0.420	0.431
	jcn	0.469	0.535
	JI	0.540	0.555

Table 4: Sentence2phrase subtask scores for the test set. *Best* indicates the best score for the subtasks. LCS stands for Normalized Longest Common Substring. Subscripts in AI-KU systems indicate the run number.

served word position (i.e., context).

The remainder of the paper proceeds as follows. Section 2 explains the preprocessing part, the difference between the systems, co-occurrence modeling, and how we calculate the similarity between two texts after co-occurrence modeling has been done. Section 3 discusses the results of our systems and makes a comparison with other participants. Section 4 concludes the findings and ends with a short future work.

2 Algorithm

In this section, we will explain preprocessing steps of the data and the details of our two participated systems[1] for this task. Both systems rely on the co-occurrence statistics. The slight difference between the first and the second system is that the first one uses the words occur in the text (e.g., paragraph, sentence etc.), whereas the second system employs co-occurrence statistics on 100 most likely substitutes of each occurred word in text.

2.1 Data Preprocessing

Two AI-KU systems can be distinguished between each other on input of co-occurrence modeling part.

AI-KU$_1$: This system uses the words that observed in the text. All words are translated into lower-case equivalents. Lemmatization[2], stop-word removal has been done and non-alphanumeric characters are excluded. Table 1 shows the input of the AI-KU$_1$ for the following sentence which is an instance from paragraph2sentence test set:

How can my blog be more popular?

AI-KU$_2$: Unlike the first system, this system represents each context of a word by finding the most likely substitutes suggested by the 4-gram language model we built from ukWaC[3] (Ferraresi et al., 2008), a 2-billion word web-gathered corpus. High probability substitutes worked successfully on Word Sense Induction (WSI) (Baskaya et al., 2013) and Part-of-Speech Induction (Yatbaz et al., 2012) problems. We used the 100 most likely substitute for each context. No lemmatization, stop word removing and lower-case transformation has been done. Table 2 illustrates the context and substitutes of each context using a bigram language model.

2.2 Co-Occurrence Modeling

This subsection will explain the unsupervised method we employed to model co-occurrence statistics: the Co-occurrence data Embedding (CODE) method (Globerson et al., 2007) and its spherical extension (S-CODE) proposed by Maron

[1]The code to replicate our work can be found *input github repo link here*.

[2]Lemmatization is carried out with Stanford CoreNLP and transforms a word into its canonical or base form.

[3]Available here: http://wacky.sslmit.unibo.it

Word	Context	Substitutes
the	<s> ___ dog	the (0.39), a (0.41), his (0.01), their (0.13), stray (0.13), ..., Alice (0.02)
dog	the ___ bites	dog (0.01), spider (0.1), flea (0.05), human (0.13), ..., the (0.01)
bites	dog ___ </s>	barks (0.03), whisperer (0.23), bites (0.04), w_4 (0.13), ..., w_n (0.01)

Table 2: Contexts when using a bigram language model

et al. (2010). Unlike our previous WSI work (Baskaya et al., 2013) where we ended up with an embedding for each word in the co-occurrence modeling step, in this task however, we try to model each text unit such as a paragraph, a sentence or a phrase, thus, after this step, we obtain embeddings for each paragraph, sentence and phrase instance.

Input data for S-CODE algorithm were instance-id of the text unit and the each word in the text unit for the first system (see Table 1), the 100 most likely substitutes of each word in text for the second system. In the initial step, S-CODE puts all instance-ids and words/substitutes randomly on n-dimensional sphere. If two different instances have the same word/substitute, then these two instances attract each other, otherwise they will repel. When S-CODE converges, instances have similar words/substitutes will be closely located, instances have no similar words/substitutes will be far from each other.

AI-KU$_1$: According to the training set performances for various n (i.e., number of dimensions for S-CODE algorithm), we picked 100 for both tasks.

AI-KU$_2$: Prior performance tests on training set, we picked n equals to 200 and 100 for paragraph2sentence and sentence2phrase subtasks, respectively.

2.3 Similarity Calculation

When the S-CODE converges, we have an n-dimensional embedding for each textual level (e.g., paragraph, sentence, phrase) instance. We can use a similarity metric to calculate the similarity between embeddings. For this task, systems should report only the similarity between two specific cross level instances. Note that we used cosine similarity in order to calculate similarity between two textual unit. *How about the other metrics? Make a test and input the reason why we use cosine similarity*. This similarity is the eventual similarity for two instances; no further processing (e.g., scaling) has been done.

3 Evaluation Results

In this task, systems were evaluated with two correlation metrics; Pearson correlation and Spearman's rank correlation. Pearson correlation tests the degree of similarity between the system's similarity ratings and the gold standard ratings. Spearman's rho tests the degree of similarity between the rankings of the items according to similarity.

Table 3 and 4 show the scores for Paragraph-2-Sentence and Sentence-2-Phrase subtasks, respectively. These tables contain the best individual scores for the performance metrics, Normalized Longest Common Substring (LCS) baseline which is given by task organizers and three additional baselines: lch (Leacock and Chodorow, 1998), jcn (Jiang and Conrath, 1997), and Jaccard Index (JI) baseline. jcn uses information content (Resnik, 1995) of the least common subsumer of concepts A and B. Information content indicates the specificity of a concept and least common subsumer of a concept A and B is the most specific concept from which A and B inherited. *Paraphrase: jcn measure augments the information content of the least common subsumer with the sum of the information content of concepts A and B themselves and takes the difference of this sum and the information content of the least common subsumer.* On the other hand, *Paraphrase: lch is based on path lengths between a pair of concepts. It finds the shortest path between two concepts, and scales that value by the maximum path length found in the isa hierarchy in which they occur* (please see Pedersen et al. (2004) for further details of these measures). These two baselines were calculated as follows. First, using the Stanford Part-of-Speech Tagger (Toutanova and Manning, 2000) we tagged words in all textual levels. After tagging, we found the synsets of each word matched with its part-of-speech using WordNet 3.0 (Fellbaum, 1998). For each synset of a word in the shorter textual unit (e.g., sentence is shorter than paragraph), we cal-

culated the jcn/lch measure of each synset of all words in the longer textual unit and picked the highest score. When we found the scores for all words, we sum them up and normalized it with the length of the shorter textual unit. This gave us the similarity between one pair in the test set. Finally, Jaccard Index baseline simply calculates the number of words in common (intersection) with two cross textual levels, normalized by the total number of words (union).

Paragraph2Sentence: Both systems outperformed all the baselines for both metrics. The best score for this subtask was .837 and our systems achieved .732 and .698 on Pearson and did similar on Spearman metric. These scores are promising since our current unsupervised systems are based on bag-of-word approach; they do not utilize any syntactic information (e.g., *input here with some examples*). Jaccard Index baseline score was .640. *Investigate why wordnet baselines are very low. Input the reason*.

Sentence2Phrase: In this subtask, AI-KU systems outperformed all baselines except that AI-KU_2 system did slightly worse than LCS on Spearman metric. Performances of systems and baselines were lower than Paragraph2Sentence subtask, since smaller textual unit (i.e., phrase) make the problem harder.

4 Conclusion

In this work, we presented two unsupervised systems that utilize co-occurrence statistics and represent textual units as a dense, low dimensional embeddings. Although current systems are based on bag-of-word approach and discard the syntactic information, they achieved promising results both paragraph2sentence and sentence2phrase subtasks. *Add some sentences to be an explanation for the prior work that didn't perform well.* As a future work, we will extend our systems by adding syntactic information *dependency parsing information? But, then we'll no longer be unsupervised?* into S-CODE.

References

Osman Baskaya, Enis Sert, Volkan Cirik, and Deniz Yuret. 2013. Ai-ku: Using substitute vectors and co-occurrence modeling for word sense induction and disambiguation. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 300–306.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2012. Learning entailment relations by global graph structure optimization. *Computational Linguistics*, 38(1):73–111.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407.

Christine Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *In Proceedings of the 4th Web as Corpus Workshop (WAC-4*.

Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. 2007. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8(10).

Aminul Islam and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(2):10.

Jay J Jiang and David W Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

C. Leacock and M. Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. In Christiane Fellfaum, editor, *MIT Press*, pages 265–283, Cambridge, Massachusetts.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78. Association for Computational Linguistics.

Yariv Maron, Michael Lamar, and Elie Bienenstock. 2010. Sphere Embedding: An Application to Part-of-Speech Induction. In J Lafferty, C K I Williams, J Shawe-Taylor, R S Zemel, and A Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1567–1575.

Eui-Kyu Park, Dong-Yul Ra, and Myung-Gil Jang. 2005. Techniques for improving web retrieval effectiveness. *Information processing & management*, 41(5):1207–1223.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2):351–383.

Kristina Toutanova and Christopher D Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics.

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

Mehmet Ali Yatbaz, Enis Sert, and Deniz Yuret. 2012. Learning syntactic categories using paradigmatic representations of word context. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 940–951. Association for Computational Linguistics.