AI-KU: Using Co-Occurrence Modeling for Semantic Similarity

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

Abstract

In this paper, we describe our unsupervised method submitted to the Cross-Level Semantic Similarity task in Semeval 2014 that computes 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 modeling step provides dense, low-dimensional embedding for each fragment which allows us to calculate semantic similarity using 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 various Natural Language Processing (NLP) tasks such as textual entailment (Berant et al., 2012), summarization (Lin and Hovy, 2003), question answering (Surdeanu et al., 2011), text classification (Sebastiani, 2002), word sense disambiguation (Schütze, 1998) and information retrieval (Park et al., 2005).

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

The Cross-Level Semantic Similarity (CLSS) task in SemEval 2014[1] (Jurgens et al., 2014) provides an evaluation framework to assess similarity methods for texts in different volumes (i.e., lexical levels). Unlike previous SemEval and *SEM tasks that were interested in comparing texts with similar volume, this task consists of four subtasks (paragraph2sentence, sentence2phrase, phrase2word and word2sense) that investigate the performance of systems based on pairs of texts of different sizes. A system should report the similarity score of a given pair, ranging from 4 (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 (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 are based on co-occurrence statistics of words. The only difference between the systems is the input they use. The 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 observed word position (i.e., context). Note that we participated two subtasks which are paragraph2sentence and sentence2phrase.

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 compares them to other participants'. Sec-

[1] http://alt.qcri.org/semeval2014/task3/

Type-ID	Lemma
Sent-33	choose
Sent-33	buy
Sent-33	gift
Sent-33	card
Sent-33	hard
Sent-33	decision

Table 1: Instance id-word pairs for a given sentence.

tion 4 discusses the findings and concludes with plans for future work.

2 Algorithm

This section explains preprocessing steps of the data and the details of our two systems[2]. Both systems rely on the co-occurrence statistics. The slight difference between the two is that the first one uses the words that occur in the given text fragment (e.g., paragraph, sentence), whereas the latter employs co-occurrence statistics on 100 substitute samples for each word within the given text fragment.

2.1 Data Preprocessing

Two AI-KU systems can be distinguished by their inputs. One uses the raw input words, whereas the other uses words' likely substitutes according to a language model.

AI-KU$_1$: This system uses the words that were in the text. All words are transformed into lowercase equivalents. Lemmatization[3] and stop-word removal were performed, and non-alphanumeric characters were excluded. Table 1 displays the pairs for the following sentence which is an instance from paragraph2sentence test set:

> "Choosing what to buy with a $35 gift card is a hard decision."

Note that the input that we used to model co-occurrence statistics consists of all such pairs for each fragment in a given subtask.

[2]The code to replicate our work can be found at `https://github.com/osmanbaskaya/semeval14-task3`.

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

AI-KU$_2$: Previously, the utilization of high probability substitutes and their co-occurrence statistics achieved notable performance on Word Sense Induction (WSI) (Baskaya et al., 2013) and Part-of-Speech Induction (Yatbaz et al., 2012) problems. AI-KU$_2$ represents each context of a word by finding the most likely 100 substitutes suggested by the 4-gram language model we built from ukWaC[4] (Ferraresi et al., 2008), a 2-billion word web-gathered corpus. Since S-CODE algorithm works with discrete input, for each context we sample 100 substitute words with replacement using their probabilities. Table 2 illustrates the context and substitutes of each context using a bigram language model. No lemmatization, stop-word removal and lowercase transformation were performed.

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

Input data for S-CODE algorithm consist of instance-id and each word in the text unit for the first system (Table 1 illustrates the pairs for only one text fragment) instance-ids and 100 substitute samples of each word in text for the second system. In the initial step, S-CODE puts all instance-ids and words (or substitutes, depending on the system) randomly on an n-dimensional sphere. If two different instances have the same word or substitute, then these two instances attract one another — otherwise they repel each other. When S-CODE converges, instances that have similar words or substitutes will be closely located or else, they will be distant 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$_2$: We picked n to be 200 and 100 for paragraph2sentence and sentence2phrase subtasks, respectively.

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

Word	Context	Substitutes
the	<s> ___ dog	The (0.12), A (0.11), If (0.02), As (0.07), Stray (0.001),..., w_n (0.02)
dog	the ___	cat (0.007), dog (0.005), animal (0.002), wolve (0.001), ..., w_n (0.01)
bites	dog ___ .	runs (0.14), bites (0.13), catches (0.04), barks (0.001), ..., w_n (0.01)

Table 2: Contexts and substitute distributions when a bigram language model is used. w and n denote an arbitrary word in the vocabulary and the vocabulary size, respectively.

	System	Pearson	Spearman
Paragraph-2-Sentence	AI-KU$_1$	0.671	0.676
	AI-KU$_2$	0.542	0.531
	LCS	0.499	0.602
	lch	0.584	0.596
	lin	0.568	0.562
	JI	0.613	0.644

Table 3: Paragraph-2-Sentence subtask scores for the training data. Subscripts in AI-KU systems specify the run number.

	System	Pearson	Spearman
Sentence-2-Phrase	AI-KU$_1$	0.607	0.568
	AI-KU$_2$	0.620	0.579
	LCS	0.500	0.582
	lch	0.484	0.491
	lin	0.492	0.470
	JI	0.465	0.465

Table 4: Sentence2phrase subtask scores for the training data.

Since this step is unsupervised, we tried to enrich the data with ukWaC, however, enrichment with ukWaC did not work well on the training data. To this end, proposed scores were obtained using only the training and the test data provided by organizers.

2.3 Similarity Calculation

When the S-CODE converges, there is 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 these embeddings. For this task, systems should report only the similarity between two specific cross level instances. Note that we used cosine similarity to calculate similarity between two textual units. This similarity is the eventual similarity for two instances; no further processing (e.g., scaling) has been done.

In this task, two correlation metrics were used to evaluate the systems: 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 rank correlation measures the degree of similarity between two rankings; similarity ratings provided by a system and the gold standard ratings.

3 Evaluation Results

Tables 3 and 4 show the scores for Paragraph-2-Sentence and Sentence-2-Phrase subtasks on the training data, respectively. These tables contain the best individual scores for the performance metrics, Normalized Longest Common Substring (LCS) baseline, which was given by task organizers, and three additional baselines: lin (Lin, 1998), lch (Leacock and Chodorow, 1998), and the Jaccard Index (JI) baseline. lin uses the information content (Resnik, 1995) of the least common subsumer of concepts A and B. Information content (IC) indicates the specificity of a concept; the least common subsumer of a concept A and B is the most specific concept from which A and B are inherited. lin similarity[5] returns the difference between two times of the IC of the least common subsumer of A and B, and the sum of IC of both concepts. On the other hand, lch is a score denoting how similar two concepts are, calculated by using the shortest path that connects the concept and the maximum depth of the taxonomy in which the concepts occur[6] (please see Pedersen et al. (2004) for further details of these measures). These two baselines were calculated as follows. First, using the Stan-

[5]lin similarity $= 2 * IC(lcs)/(IC(A) + IC(B))$ where lcs indicates the least common subsumer of concepts A and B.

[6]The exact formulation is $-log(L/2d)$ where L is the shortest path length and d is the taxonomy depth.

	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	0.629	0.627
	lin	0.612	0.601
	JI	0.640	0.687

Table 5: Paragraph-2-Sentence subtask scores for the test data. *Best* indicates the best correlation score for the subtask. LCS stands for Normalized Longest Common Substring. Subscripts in AI-KU systems specify 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	0.526	0.544
	lin	0.501	0.498
	JI	0.540	0.555

Table 6: Sentence2phrase subtask scores for the test data.

ford Part-of-Speech Tagger (Toutanova and Manning, 2000) we tagged words across 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 calculated the lin/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 calculated the mean to find out the similarity between one pair in the test set. Finally, Jaccard Index baseline was used to simply calculate the number of words in common (intersection) with two cross textual levels, normalized by the total number of words (union). Table 5 and 6 demonstrate the AI-KU runs on the test data. Next, we present our results pertaining to the test data.

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-words approach — they do not utilize any syntactic information.

Sentence2Phrase: In this subtask, AI-KU systems outperformed all baselines with the exception of the AI-KU$_2$ system which performed slightly worse than LCS on Spearman metric. Performances of systems and baselines were lower than Paragraph2Sentence subtask, since smaller textual units (such as phrases) make the problem more difficult.

4 Conclusion

In this work, we introduced two unsupervised systems that utilize co-occurrence statistics and represent textual units as dense, low dimensional embeddings. Although current systems are based on bag-of-word approach and discard the syntactic information, they achieved promising results in both paragraph2sentence and sentence2phrase subtasks. For future work, we will extend our algorithm by adding syntactic information (e.g, dependency parsing output) into the co-occurrence modeling step.

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.

David Jurgens, Mohammed Taher Pilehvar, and Roberto Navigli. 2014. Semeval-2014 task 3: Cross-level semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014). August 23-24, 2014, Dublin, Ireland.*

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.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304.

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.