Duluth : Measuring Cross–Level Semantic Similarity with First and Second–Order Dictionary Overlaps

Ted Pedersen
Department of Computer Science
University of Minnesota
Duluth, MN 55812
tpederse@d.umn.edu

Abstract

This paper describes the Duluth systems that participated in the Cross–Level Semantic Similarity task of SemEval–2014. These three systems were all unsupervised and relied on a dictionary melded together from various sources, and used first–order (Lesk) and second–order (Vector) overlaps to measure similarity. The first–order overlaps fared well according to Spearman's correlation (top 5) but less so relative to Pearson's. Most systems performed at comparable levels for both Spearman's and Pearson's measure, which suggests the Duluth approach is potentially unique among the participating systems.

1 Introduction

Cross–Level Semantic Similarity (CLSS) is a novel variation on the problem of semantic similarity. As traditionally formulated, pairs of words, pairs of phrases, or pairs of sentences are scored for similarity. However, the CLSS shared task (Jurgens et al., 2014) included 4 subtasks where pairs of different granularity were measured for semantic similarity. These included : word-2-sense (w2s), phrase-2-word (p2w), sentence-2-phrase (s2p), and paragraph-2-sentence (g2s). In addition to different levels of granularity, these pairs included slang, jargon and other examples of non–standard English.

We were drawn to this task because of our long–standing interest in semantic similarity. We have pursued approaches ranging from those that rely on structured knowledge sources like WordNet (e.g., WordNet::Similarity) (Pedersen et al., 2004) to those that use distributional information found in raw text (e.g., SenseClusters) (Purandare and Pedersen, 2004). Our approach in this shared task is a bit of both, but relies on using definitions for each item in a pair so that similarity can be measured using first or second–order overlaps.

A first–order approach finds direct matches between the words in a pair of definitions. In a second–order approach each word in a definition is replaced by a vector of the words it co–occurs with, and then the vectors for all the words in a definition are averaged together to represent the definition. Then, similarity can be measured by finding the cosine between pairs of these vectors. We decided on a definition based approach since it had the potential to normalize the differences in granularity of the pairs.

The main difficulty in comparing definitions is that they can be very brief or may not even exist at all. This is why we combined various different kinds of resources to arrive at our dictionary. While we achieved near total coverage of words and senses, phrases were sparsely covered, and sentences and paragraphs had no coverage. In those cases we used the text of the phrase, sentence or paragraph to serve as its own definition.

The Duluth systems were implemented using the UMLS::Similarity package (McInnes et al., 2009) (version 1.35)[1], which includes support for user–defined dictionaries, first–order Lesk methods, and second–order Vector methods. As a result the Duluth systems required minimal implementation, so once a dictionary was ready experiments could begin immediately.

This paper is organized as follows. First, the first–order Lesk and second–order Vector measures are described. Then we discuss the details of the three Duluth systems that participated in this task. Finally, we review the task results and consider future directions for this problem and our system.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Page numbers and proceedings footer are added by the organisers. Licence details: http://creativecommons.org/licenses/by/4.0/

[1] http://umls-similarity.sourceforge.net

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 247–251,
Dublin, Ireland, August 23-24 2014.

2 Measures

The Duluth systems use first–order Lesk methods (Duluth1 and Duluth3) and second–order Vector methods (Duluth2). These require that definitions be available for both items in a pair, with the caveat that we use the term *definition* somewhat loosely to mean both traditional dictionary definitions as well as various proxies when those are not available.

2.1 First–order Overlaps : Lesk

The Lesk measure (Lesk, 1986) was originally a method of word sense disambiguation that measured the overlap among the definitions of the possible senses of an ambiguous word with those of surrounding words (Lesk, 1986). The senses which have the largest number of overlaps are presumed to be the correct or intended senses for the given context. A modified approach compares the glosses of an ambiguous word with the surrounding context (Kilgarriff and Rosenzweig, 2000). These are both first–order methods where definitions are directly compared with each other, or with the surrounding context.

In the Duluth systems, we measure overlaps by summing the number of words shared between definitions. Sequences of words that match are weighted more heavily and contribute the square of their length, while individual matching words just count as one. For example, given the definitions *a small noisy collie* and *a small noisy border collie* the stop word *a* would not be matched, and then *small noisy* would match (and be given a score of 4) and then *collie* would also match (receiving a score of 1). So, the total Lesk score would be 5. The scores of the Duluth systems were normalized by dividing by the maximum Lesk score for any pair in a subtask. This moves the scores to a 0–1 scale, where 1.00 means the definitions are exactly the same, and where 0 means they share no words.

One of the main drawbacks of the original Lesk method is that glosses tend to be very short. Various methods have been proposed to overcome this. For example, (Banerjee and Pedersen, 2003) introduced the Extended Gloss Overlap measure which creates super–glosses by augmenting the glosses of the senses to be measured with the glosses of semantically related senses (which are connected via relation links in WordNet). This adaptation of the Lesk measure was first implemented in WordNet::Similarity (Pedersen et al., 2004) and then later in UMLS::Similarity (McInnes et al., 2009). It has been applied to both word sense disambiguation and semantic similarity, and generally found to improve on original Lesk (Banerjee, 2002; Banerjee and Pedersen, 2002; Patwardhan et al., 2003; McInnes and Pedersen, 2013). However, the Duluth systems do not build super–glosses in this way since many of the items in the pairs are not found in WordNet. However, definitions are expanded in a simpler way, by merging together various different resources to increase both coverage and the length of definitions.

2.2 Second–order Overlaps : Vector

The main limitation of first–order Lesk approaches is that if terminology differs from one definition to another, then meaningful matches may not be found. For example, consider the definitions *a small noisy collie* and *a dog that barks a lot*. A first–order overlap approach would find no similarity (other than the stop word *a*) between these definitions.

In cases like this some form of term expansion could improve the chances of matching. Synonym expansion is a well–known possibility, although in the Duluth systems we opted to expand words with their co–occurrence vectors. This follows from an approach to word sense discrimination developed by (Schütze, 1998). Once words are expanded then all the vectors in a definition are averaged together and this averaged vector becomes the representation of the definition. This idea was first implemented in WordNet::Similarity (Pedersen et al., 2004) and then later in UMLS::Similarity (McInnes et al., 2009), and has been applied to word sense disambiguation and semantic similarity (Patwardhan, 2003; Patwardhan and Pedersen, 2006; Liu et al., 2012).

The co–occurrences for the words in the definitions can come from any corpus of text. Once a co–occurrence matrix is constructed, then each word in each definition is replaced by its vector from that matrix. If no such vector is found the word is removed from the definition. Then, all the vectors representing a definition are averaged together, and this vector is used to measure against other vectors created in the same way. The scores returned by the Vector measure are between 0 and 1 (inclusive) where 1.00 means exactly the same and 0 means no similarity.

3 Duluth Systems

There were three Duluth systems. Duluth1 and Duluth3 use first–order Lesk, and Duluth2 uses second–order Vector. Duluth3 was an ensemble made up of Duluth1 and a close variant of it (Duluth1a, where the only difference was the stop list employed).

Duluth1 and Duluth2 use the NSP stoplist[2] which includes approximately 390 words and comes from the SMART stoplist. Duluth1a treated any word with 4 or fewer characters as a stop word. Stemming was performed by all Duluth systems using the Porter algorithm as implemented in the Lingua::Stem::en Perl module.

Before processing, all of the similarity pairs and the dictionary entries were converted to lower case and any non alpha-numeric characters were replaced with spaces. Also, any stop listed words were removed.

3.1 Dictionary Creation

The key step for all the Duluth systems is the creation of the dictionary. We elected to treat senses as word forms, and so our dictionary did not make sense distinctions (and would include all the senses of a word or phrase in its entry).

Since the words and phrases used in some pairs are slang or non–standard English, traditional lexical resources like WordNet do not provide adequate coverage. However, WordNet provides a good foundation for coverage of standard English, so we began by extracting the glosses from WordNet v3.0 using the WordNet::QueryData Perl module.

Wiktionary is a crowd sourced lexical resource that includes more slang and jargon, so we also extracted entries from it using the Wiktionary::Parser Perl module. In hopes of increasing our coverage of phrases in particular, we looked up words and phrases in Wikipedia using the WWW::Wikipedia Perl module and used the first paragraph of an entry (up to the first heading) as a definition. Finally, we also used the `dict` program in Linux which we configured to use the following resources : the Collaborative International Dictionary of English v.0.48 (gcide), Moby Thesaurus II by Grady Ward, 1.0 (moby-thes), V.E.R.A. – Virtual Entity of Relevant Acronyms (June 2006) (vera), the Jargon File (version 4.4.7, 29 Dec 2003) (argon), the Free On-line Dictionary of Computing (26 July 2010) (foldoc), and the CIA World Factbook 2002 (world02).

The most obvious question that arises about these resources is how much coverage they provide for the pairs in the task. Based on experiments on the trial data, we found that none of the resources individually provided satisfactory coverage, but if they were all combined then coverage was reasonably good (although still not complete). In the test data, it turned out there were only 20 items in the w2s subtask for which we did not have a dictionary entry (out of 1000). However, for p2w (phrase-2-word) there were 407 items not included in the dictionary (most of which were phrases). In the s2p (sentence-2-phrase) subtask there were only 15 phrases which had definitions, so for this subtask and also for g2s (paragraph-2-sentence) the items themselves were the definitions for essentially all the pairs.

Also of interest might be the total size of the dictionaries created. The number of tokens in g2s (paragraph-2-sentence) was 46,252, and in s2p (sentence-2-phrase) it was 12,361. This is simply the token count for the pairs included in each subtask. However, the dictionaries were much larger for p2w (phrase-2-word), where the token count was 262,876, and for w2s (word-2-sense) where it was 499,767.

3.2 Co–occurrence Matrix for Vector

In the Duluth systems, the co–occurrence matrix comes from treating the WordNet glosses as a corpus. Any pair of words that occur together in a WordNet gloss are considered a co–occurrence.

There are 117,659 glosses, made up of 1,460,921 words. This resulted in a matrix of 90,516 rows and 99,493 columns, representing 708,152 unique bigrams. The matrix is not symmetric since the co–occurrences are bigrams, so *dog house* is treated differently than *house dog*.

The WordNet glosses were extracted from version 3.0 using the glossExtract Perl program[3].

4 Results

Results for the CLSS task were ranked both by Pearson's and Spearman's Correlation coefficients. Duluth system results are shown in Tables 1 and 2. These tables also include the results of

[2] http://cpansearch.perl.org/src/TPEDERSE/Text-NSP-1.27/bin/utils/stoplist-nsp.regex

[3] http://www.d.umn.edu/~tpederse/Code/glossExtract-v0.03.tar.gz

Table 1: Spearman's Results

	g2s	s2p	p2w	w2s	rank (of 38)
Top	.801	.728	.424	.343	1
Duluth3	.725	.660	.399	.327	3
Duluth1	.726	.658	.385	.316	5
Duluth2	.553	.473	.235	.231	21
Baseline	.613	.626	.162	.128	

Table 2: Pearson's Results

	g2s	s2p	p2w	w2s	rank (of 38)
Top	.811	.742	.415	.355	1
Duluth2	.501	.450	.241	.224	23
Duluth1	.458	.440	.075	.076	30
Duluth3	.455	.426	.075	.080	31
Baseline	.527	.562	.165	.110	

the top ranked system (which was the same system according to both measures) and results from a baseline system that measures the Least Common Substring between the terms in a pair, except in the w2s subtask, where it measured the LCS between the associated WordNet glosses.

Table 1 shows that the Duluth3 system offers a slight improvement upon Duluth1. Recall that Duluth3 is an ensemble that includes Duluth1 and its minor variant Duluth1a. Both of these are first–order methods, and significantly outperform the second–order method Duluth2.

However, Table 2 tells a completely different story. There the second–order system Duluth2 performs better, although overall rankings suffer according to Pearson's measure. It is also very apparent that the ranks between Pearson's and Spearman's for Duluth1 and Duluth3 differ significantly (from 3 to 30 and 5 to 31). This is very atypical, and most systems maintained approximately the same rankings between the two correlation measures. Note that Duluth2 behaves in this way, where the relative ranking is 21 and 23.

Table 3 shows the number of pairs in each subtask which returned a score of 0. This could be due to missing definitions, or no matches occurring between the definitions. Interestingly Duluth2 has a much smaller number of 0 valued scores, which shows the second–order method provides greater coverage due to its more flexible notion of matching. However, despite much higher numbers of

Table 3: Number of Pairs with Score of 0

	g2s	s2p	p2w	w2s
Duluth1	107	197	211	23
Duluth2	9	101	40	15
Duluth3	101	196	205	23

0s, Duluth1 and Duluth3 perform much better with Spearman's rank correlation coefficient. This suggests that there is a kind of precision–recall tradeoff between these systems, where Duluth2 has higher recall and Duluth1 and Duluth3 have higher precision.

5 Future Directions

The relatively good performance of the first–order Duluth systems (at least with respect to rank correlation) shows again the important role of lexical resources. Our first–order method was not appreciably more complex than the baseline method, yet it performed significantly better (especially for the p2w and w2s tasks). This is no doubt due to the more extensive dictionary that we employed.

That said, our approach to building the dictionary was relatively crude, and could be substantially improved. For example, we could be more selective in the content we add to the entries for words or phrases. We could also do more than simply use the sentences and paragraphs as their own definitions. For example, we could replace words or phrases in sentences and paragraphs with their definitions, and then carry out first or second–order matching.

Second–order matching did not perform as well as we had hoped. We believe this is due to the somewhat noisy nature of the dictionaries we constructed, and expanding those definitions by replacing words with vectors created even more noise. We believe that a more refined approach to creating dictionaries would certainly improve these results, as would a more selective method of combining the co–occurrence vectors (rather than simply averaging them).

Acknowledgments

The Duluth systems relied heavily on the freely available software package UMLS::Similarity. We are grateful to Bridget McInnes and Ying Liu for their work in developing this package, and in particular for the `--dict` functionality.

References

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, pages 136–145, Mexico City, February.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, Acapulco, August.

Satanjeev Banerjee. 2002. Adapting the Lesk algorithm for word sense disambiguation to WordNet. Master's thesis, University of Minnesota, Duluth, December.

David Jurgens, Mohammad 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)*, Dublin, August.

Adam Kilgarriff and Joseph Rosenzweig. 2000. Special issue on SENSEVAL: Framework and results for english SENSEVAL. *Computers and the Humanities*, 34(1–2):15–48.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM Press.

Ying Liu, Bridget McInnes, Ted Pedersen, Genevieve Melton-Meaux, and Serguei Pakhomov. 2012. Semantic relatedness study using second–order co–occurrence vectors computed from biomedical corpora, UMLS, and WordNet. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 363–371, Miami, FL.

Bridget McInnes and Ted Pedersen. 2013. Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text. *Journal of Biomedical Informatics*, 46:1116–1124.

Bridget McInnes, Ted Pedersen, and Serguei Pakhomov. 2009. UMLS-Interface and UMLS-Similarity : Open source software for measuring paths and semantic similarity. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 431–435, San Francisco.

Siddharth Patwardhan and Ted Pedersen. 2006. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *Proceedings of the EACL 2006 Workshop on Making Sense of Sense: Bringing Computational Linguistics and Psycholinguistics Together*, pages 1–8, Trento, Italy, April.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257, Mexico City, February.

Siddharth Patwardhan. 2003. Incorporating dictionary and corpus information into a context vector measure of semantic relatedness. Master's thesis, University of Minnesota, Duluth, August.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::Similarity - Measuring the relatedness of concepts. In *Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 38–41, Boston, MA.

Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 41–48, Boston, MA.

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