SynId: A hybrid approach to semantic word similarity for use in Automated Essay Scoring (AES) software

Jeremy Dohmann[1], Patrick DeMichele[2], and Michael Lepori[3]

[1]Harvard University
[2]Stanford University
[3]Johns Hopkins University

In this paper, we discuss the development of a semantic word similarity algorithm which relies on an ensemble of corpus-based syntactic and contextual measures as well as part-of-speech information in order to perform a binary classification task designed for use in AES software. Our definition of semantic similarity was tailored to the constraints of the task, thus training and testing of the classifier was performed on hand-tagged data consistent with our definition. Using a set of 17 features and a random forest classifier, we achieved an accuracy of 84% and a Cohen's Kappa of .67 indicating substantial agreement with the human-assigned tags. Since our data was generated specifically for this task, in order to compare against an industry benchmark, we compared our algorithm's performance to the performance of a number of measures offered by the JWI-2.4.0 package.

1 Introduction

Semantic "similarity" is not a well-defined term. It can be variously be taken to mean synonymy/antonymy, hypernymy/hyponymy, contextual relatedness, topical relatedness, or some combination thereof. In fact, the processes which humans undergo when they judge words and phrases as similar are not entirely clear. Though our mental representation of linguistic knowledge is often considered to take the form of an associative network where lexical activation is spread from adjacent concepts, the exact underlying representation is elusive as is the mechanism by which context and world knowledge mediate this lexical priming.

Despite the nuances of this subject, calculating text similarity has become a task of fundamental importance in many Natural Language Processing tasks including information retrieval, question answering, text classification, document clustering, and automated essay scoring. Measuring the similarity of words is often the first step in comparing the similarity of sentences, paragraphs, or entire documents. Common methods such as latent semantic analysis (LSA), latent dirichlet allocation (LDA), or word embeddings often consider texts using a bag-of-words model whereby a passage is represented as the vector sum of the words it contains.

Broadly-speaking, semantic word similarity measures can be grouped into three categories: corpus-based measures, ontology or knowledge-based measures, and hybrid measures which combine strategies from the prior two categories[5]. Ontology-based measures can be highly effective using narrow definitions of relatedness, especially in cases where similarity is gauged via relations coded explicitly into the knowledge-base (e.g. hypyernymy, synonymy). When using broader definitions based on topicality or contextual similarity, they are often outperformed by corpus-based measures such as LSA and LDA, especially when used to compare semantic similarity at the sentence or document level[3]. Furthermore, they suffer from prohibitive development costs. WordNet, the most popularly used semantic knowledge-base, contains 155,287 words organized in 117,659 synsets for a total of 206,941 word-sense pairs all of which were hand-coded[1].

Corpus-based measures, however, are highly robust to broad definitions of similarity as they are capable of leveraging implied semantic associations mined from troves of contextual data. As an example, consider the four sentences below:

- The dog is the boy's pet.
- Cats make good pets.
- Dogs are furry.
- Cats are furry.

In the above example, "dog" and "cat" do not appear in the same context. That said, however, they appear in contexts that are similar to one another, in fact they are linked to one another via their mutual association to the words "furry" and "pet/pets". This inductive learning of word associations is highly typical for corpus-based approaches such as LSA[6]. One major weakness of corpus-based measures, especially LSA, is

that they demand tremendous computational resources both in terms of storage of corpora and generation of vector representations.

In addition to corpus-based and ontology-based measures, hybrid measures have been developed in order to mitigate the weaknesses of the different approaches. In [7], two corpus-based and six ontology-based similarity measures were analyzed and it was shown that a combination of several different measures made for the best results.

2 Methods

2.1 Definition of similarity

This word similarity classifier was developed for use in an Automated Essay Scoring (AES) system with the intention of using it to identify textual cohesion within the essay as well as relevance of the text to the prompt and source material provided. To that end, our definition of semantic similarity embraces concerns regarding textual cohesion and topicality. Thus we deem word pairs to be similar not only if they are related to one another by synonymy, hypernymy, etc. but also if they share common topicality or if the presence of pairs of words in adjacent sentences of an essay would indicate textual cohesion. Additionally, we only consider word pairs consisting of adjectives, nouns, verbs, adverbs, or some combination thereof.

To demonstrate what this definition entails, several positive and negative examples taken from our training data have been provided below

Positive

- Equivoca/JJ /Uncertain/JJ
- Brioche/NN Brunch/NN
- Proposal/NN Marriage/NN
- Bed/NN Pillow/NN

Negative

- Public/NN Subject/NN
- Large/JJ Structure/NN
- Liver/NN General/NN
- Knowledge/NN Abstract/JJ

As shown, positive examples consist of synonyms (e.g. equivocal/uncertain), topically related concepts (e.g. proposal/marriage) or items which are highly related to each other temporally, spatially or otherwise (e.g. bed/pillow, brioche/brunch). Negative examples, additionally, can be completely unrelated, tangentially related (e.g. knowledge/abstract) or common bigrams (e.g. large/structure, knowledge/abstract) which aren't necessarily topically related to one another. Examples such as large/structure and knowledge/abstract are particularly thorny because common bigrams often are semantically related, but mere high incidence of co-occurrence is not enough to qualify a pair as semantically related. That said, it would not be unreasonable for one to make the case that some of the negative pairs we've provided should be classified as positive examples.

The ambiguity of our examples is emblematic of the ambiguity of the task at hand. Though the consequent decision function is nuanced and oftentimes ambiguous, it very clearly reflects the character of the task it sets out to solve. Though such ambiguity presents a classification task of tremendous difficulty, the payoff is significant as it enables the development of a robust and expressive classifier capable of identifying the subtle contextual, syntactic, and topical similarities which tie words together.

2.2 Features

Our classifier depends upon a set of 17 features, all of which generate real-valued numbers, which are then fed into a random forest classifier consisting of 300 trees. Our features fall into three primary categories: syntactic, contextual, and miscellaneous. In order to generate our features, two graph representations of our corpora were constructed, one reflecting the syntactic connectivity of the words in our corpus, and another reflecting the contextual connectivity of the vocabulary.

In our syntactic connectivity graph, each word represented a node, and the weights connecting different nodes was proportional to the number of times in the corpus the two words had been connected to one another via a grammatical dependency as extracted using Stanford CoreNLP. The adjacency list of each node was separated into 9 lists according to the part-of-speech of the adjacent nodes. In our contextual connectivity graph, our corpora were segmented into three-sentence-wide windows and weights between words were made proportional to the number of times two words appeared in the same window.

2.2.1 Syntactic features

The syntactic features used were of two basic types: adjacency list overlap, and two-tiered adjacency list overlap.

Adjacency list overlap: This category consists of 8 features measuring the overlap of the word pairs' adjacency lists for 8 different part of speech categories (all except pronouns). Though not a direct probability, the features are proportional to the probability that a word occurring in word 1's adjacency lists occurs in word 2's corresponding list. For example, if the given word pair were Cat/Dog and **fuzzy** occurred in the adjective adjacency lists for both words with high frequency then it would contribute to a higher value for the corresponding feature. Analogously, if **dogs** and **cats** were generally described using similar adjectives, then the corresponding feature would have a high value.

Two-tiered adjacency list overlap: This category consists of 2 features measuring the overlap of the adjacency lists of the adjacency lists of the word pairs for two different part of speech categories (NN-NN and NN-VB). The features are directly proportional to the probability that a word occurring in one of the adjacency lists in word 1's adjacency list also occurs in the adjacency list of a word in word 2's adjacency lists, where the list combinations explored are specified by the two categories above. For example, if the given word pair were Cat/Dog and **owner** were listed in the NN adjacency list of **cat** and **pet** were listed in the VB adjacency list of **owner** and **man** were in **dog**'s NN list and **pet** in the VB list of **man** then the fact that nouns associate with Cat/Dog respectively engage in the mutual activity of petting would contribute to a higher value for the corresponding feature. Analogously, if **dogs** and **cats** are generally associated with nouns that perform similar actions, or nouns that are associated with similar nouns then the corresponding features would have high values.

2.2.2 Contextual features

Four contextual features were used: adjacency list overlap, two-step path distance, explicit *tf-idf*, and two-tiered adjacency list overlap.

Adjacency list overlap: Analogously to the syntactic equivalent of this feature, this feature measures the overlap of words in the adjacency lists of the words in the pair. The primary differences, however, are the that there is no differentiation according to part of speech, and that there is a *tf-idf* cutoff below which words are not counted towards the overlap. All words adjacent to a given word with a *tf-idf* (with respect to the given word) below 50 are skipped when calculating this feature.

Two-step path distance: For this feature, two-step paths are considered those for which word 2 is in the adjacency list of a word in word 1's adjacency list. The distance for said path is taken to be the geomeasure mean of the *tf-idf*'s of the connecting word with respect to words 1 and 2. The feature is the arithmetic mean of all such paths existing between the two words in the pair.

Explicit *tf-idf*: This feature measures the *tf-idf* of word 1 with respect to word 2 and vice versa then returns the arithmetic mean of the two values.

Two-tiered adjacency list overlap: Analogously to the syntactic equivalent of this feature, the two-tiered adjacency list overlap measures the overlap of words in the adjacency lists of the words in the adjacency lists of the words in the pair. Just like above, however, there is no differentiation between parts of speech, and there is a *tf-idf* cutoff of 50 below which words do not contribute to the value returned.

2.2.3 Miscellaneous features

There are three miscellaneous features: string overlap, WordNet definition overlap, and part of speech class.

String overlap: String overlap measures the length of the longest shared character sequence divided by the length of the shortest word in the pair. For example **love** and **loving** would return a value of .75.

WordNet definition overlap: For this feature, the definitions for the primary synset associated with the words in WordNet are generated and purified of stop words and then the percentage of shared words is calculated and returned.

POS-class: This feature returns an integer between 1 and 10 corresponding to the 10 possible POS combinations. Since there are 4 parts-of-speech considered there are 10 possible combinations when sampled with replacement (NN-NN, NN-VB, NN-RB, NN-JJ, etc.).

2.3 Data

2.3.1 Corpora

Our corpora were sourced from several broad genres of English text, including such categories as news, law, literature, poetry, encyclopedia, and spoken word. In total, our corpus contains approximately 27.5 million words, with the most well represented domains being literature, spoken word, and encyclopedia. Individual domain-specific corpora were created either by employing a web crawler to extract publicly available texts, or through manually downloading and refining texts from open-source works.

2.3.2 Word pairs

The negative training and testing examples were generated by randomly selecting pairs of words from our corpus, filtering out words of improper POS and tagging those which were truly negative examples. The positive examples were generated through a more involved process by which the first word was randomly selected. Human specialists were then presented each first word and prompted to provide the most relevant word which came to mind. This process may have biased selection of words which were common bigrams but participants were instructed to consider as many possible pair words as they could before settling on the most relevant one. In order to mitigate the biases of a particular individual, several individuals contributed in tagging the words.

3 Results and Discussion

Our features, being that they are largely corpus-based measures, are relatively expensive to compute. The source files containing the graph representations of our data are 2GB in total. Loading in the two graph models takes approximately 100s on a System76 Linux workstation containing 64GB of RAM and a 4.4GHz processor. It is likely that this speed could be improved using multithreading. Further generating the vector for each word pair takes roughly .5s per word pair, though this number can be as low as .005 or as high as 3-4s depending on the frequency of the respective words in the corpora.

3.1 Performance

The classifier was trained using 916 +/- examples generated according to the procedure in section **2.3.2** and tested on a set of 152 positive and 158 negative examples drawn from the same distribution. The accuracy was 83.87% and the Cohen's Kappa interrater agreement measure was .669 indicating substantial agreement above chance[8]. The classifier had 126 true positives, 26 false negatives, 134 true negatives, and 24 false negatives, yielding a precision of .84, a recall of .828, and an F-measure of .834.

Several example misclassifications have been reproduced below:

False positive

1. Immortality/NN Christian/JJ
2. Beaches/NNS Turtle/NN
3. Name/NN Word/NN
4. Multiple/JJ Single/JJ
5. Innovations/NNS Writing/NN
6. Sikh/NN Provides/VBZ
7. Ice/NN Disparagement/NN
8. Pavements/NNS Pot/NN

False negative

1. Predominant/JJ Popular/JJ
2. Entertainment/NN Show/NN
3. Suit/NN Clothing/NN
4. Bandaging/VBG Nurse/NN
5. Shell/NN Beach/NN
6. Cold/JJ Ice/NN
7. Be/VB Is/VBZ
8. Capacity/NN Maximum/NN

Of the false positives, some of the examples were debatably negative and perhaps should be considered as true positives (e.g. **2**, **3**, and **4**). While others were certainly related to one another but were only tenuously connected (e.g. **1** and **5**), while others still were completely unrelated to one another (e.g.**7** and **8**). A partial explanation of the false positive data is that for those which are tangentially related but not quite semantically similar, the algorithm has a poor time disambiguating truly meaningful contextual or syntactic similarity from vacuous co-occurrence (perhaps because the difference between the two is not well-defined), and that for those which are entirely unrelated, data sparsity makes it so that a number of unlikely co-occurrences poison the data leading to misleading results. Data sparsity playing a significant role in the classifier's performance should come as no surprise especially when one considers that our corpus was 30 million words, while common corpus-based methods frequently rely on hundreds of millions of words of text[3].

The false negatives, on the other hand are more difficult to account for, though they generally fall into one of several categories. There are direct synonyms (e.g. **1**), type-of relations (e.g. **2** and **3**), actions performed by entities (e.g. **4**), associated objects (e.g. **5**), common descriptions of objects e.g. **6**), conjugations of identical verb forms (e.g. **7**), and common bigrams (e.g. **8**). The same reasons given above likely prevail for false negatives as well. No doubt performance in this respect would be improved through the use of larger corpora. Likewise, examples such as **5** and **8** once again suffer from the uncertain distinction between appropriate positive and negative examples. Furthermore, examples such as **7** are particularly difficult to handle as they likely arise from

the algorithm's dependence on explicit overlap in words' adjacency lists. Different subjects will be associated with the different verb forms making it incredibly difficult to meaningfully draw comparisons between the two words.

In general, however, the classifier has proven to be incredibly effective despite data sparsity and ill-defined decision boundaries, generating a highly robust and broad decision function capable of making subtle inferences regarding the relationships of words.

3.2 Benchmarks

In order to contextualize our performance with respect to industry standards, we use three other popular similarity techniques taken from the JWI-2.4.0 package [9]. These measures are the Jiang and Conrath measure [4], the Lin measure [10], and the JWSRandom measure imported directly from JWI, authored by David Hope. The three measures are hybrid corpus/ontology-based measures which rely on information content regarding the concepts subsuming the two words in a knowledge-base. Jiang and Conrath and Lin were both described in a 2013 metanalysis on semantic similarity measures and compared against other common measures [2].

Of the 14 measures compared on a common evaluation measure taken from [11], Lin was was tied for second in performance with an accuracy of 82% and Jiang and Conrath was tied for tenth with an accuracy of 73%.

On the present dataset, the three measures given above were used as features into a random forest classifier of size 300. The features were formed into every possible combination and tested on the dataset 100 times. The most effective pairing, ranked according to Cohen's Kappa, consisted of the Lin and Jiang and Conrath measures combined. Using this combination, the random forest boasted an average accuracy of 58.71% and Cohen's Kappa of 0.186.

SynId far out-performed that based on the three measures given above, though the Jiang and Conrath and Lin classifier was 10 times faster than the classifier currently presented. Additionally, SynId does not require words be contained in WordNet in order to correctly classify them, and instead requires merely that the words occur at least once in the corpora.

4 Conclusion

We have presented SynId, a hybrid approach to semantic similarity measurement which relies on statistical syntactic and contextual information gleaned from large, natural text corpora. SynId's performance outcompetes several industry standard similarity measures on the particular classification task presented above.

Though SynId is relatively slow compared to other measures, and still suffers from an inability to clearly distinguish the decision boundaries presented by the task, it shows great promise for use in AES software, and is capable of adopting robust similarity standards highly-suited for that task.

The authors would like to further investigate SynId's performance on other tasks related to text cohesion, namely coarser-grained text similarity tasks such as sentence-sentence, and document-document similarity measurement.

References

[1] *"WordNet Statistics"*. WNSTATS(7WN) Manual Page. Princeton University, n.d. Web. 11 Aug. 2016.

[2] Thabet Slimani. *"Description and Evaluation of Semantic Similarity Measures Approaches."*. International Journal of Computer Applications IJCA 80.10 (2013): 25-33. Arxiv.org. Web. 11 Aug. 2016.

[3] Evgeniy Gabrilovich and Shaul Markovitch *"Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis."* IJCAI-07 (2007): 1606-611. Print.

[4] Jay J. Jiang and David W. Conrath *Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy* Proceedings of International Conference Research on Computational Linguistics (ROCLING X), 1997, Taiwan

[5] Wael H. Gomaa and Aly A. Fahmy. *A Survey of Text Similarity Approaches* International Journal of Computer Applications IJCA 68.13 (2013): 13-18. Web.

[6] Thomas K. Landauer and Susan T. Dumais *A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge* Psychological Review 104.2 (1997): 211-40. Web.

[7] Rada Mihalcea, Courtney Corley and Carlo Strapparava *Corpus-based and Knowledge-based Measures of Text Semantic Similarity* UNT Digital Library. http://digital.library.unt.edu/ark:/67531/metadc30981/. Accessed August 11, 2016.

[8] Anthony J. Viera, MD; Joanne M. Garrett, PhD *Understanding Interobserver Agreement: The Kappa Statistic* Fam Med 2005;37(5):360-3.

[9] Mark Alan Finlayson *Java Libraries for Accessing the Princeton Wordnet: Comparison and Evaluation* JWI-2.4.0. http://projects.csail.mit.edu/jwi/. Web. Accessed August 11, 2016.

[10] Dekang Lin *An Information-Theoretic Definition of Similarity* http://disi.unitn.it/ p2p/RelatedWork/Matching/an-information-theoretic-definition.pdf. Web. Accessed August 12, 2016.

[11] Miller, G. A. and Charles, W. G. *Contextual correlates of semantic similarity.* Language and Cognitive Processes, 6, 1-28. 1991.