Using the Wiktionary Graph Structure for Synonym Detection

Timothy Weale, Chris Brew, Eric Fosler-Lussier
Department of Computer Science and Engineering
The Ohio State University
{weale,cbrew,fosler}@cse.ohio-state.edu

Abstract

This paper presents our work on using the graph structure of Wiktionary for synonym detection. We implement semantic relatedness metrics using both a direct measure of information flow on the graph and a comparison of the list of vertices found to be "close" to a given vertex. Our algorithms, evaluated on ESL 50, TOEFL 80 and RDWP 300 data sets, perform better than or comparable to existing semantic relatedness measures.

1 Introduction

The recent creation of large-scale, collaboratively constructed semantic resources provides researchers with cheap, easily accessible information. Previous metrics used for synonym detection had to be built using co-occurrence statistics of collected corpora (Higgins, 2004) or expensive, expert-created resources such as WordNet or Roget's Thesaurus (Jarmasz and Szpakowicz, 2003). Here, we evaluate the effectiveness of Wiktionary, a collaboratively constructed resource, as a source of semantic relatedness information for the synonym detection problem.

Researching these metrics is important because they have been empirically shown to improve performance in a variety of NLP applications, including word sense disambiguation (Turdakov and Velikhov, 2008), real-world spelling errors (Budanitsky and Hirst, 2006) and coreference resolution (Strube and Ponzetto, 2006).

Synonym detection is a recognized testbed for comparing semantic relatedness metrics (e.g (Zesch et al., 2008)). In this task, a target word or phrase is presented to the system, which is then presented with four alternative words or phrases. The goal of the system is to pick the alternative most related to the target. Example questions can be found in Figure 1.

Through the Wikimedia Foundation,[1] volunteers have created two large-scale, collaborative resources that have been used in previous relatedness research – Wikipedia (an encyclopedia) and Wiktionary (a dictionary). These sources have been used for synonym detection and replicating human relatedness evaluations using the category structure (Strube and Ponzetto, 2006), local link structure (Milne and Witten, 2008) and (Turdakov and Velikhov, 2008) and global features (Zesch et al., 2008). They contain related information but focus on different information needs; which information source provides better results depends on the needs of the task. We use Wiktionary which, due to its role as a dictionary, focuses on common words and definitions – the type of information found in our synonym detection problems.

Both Wikipedia and Wiktionary are organized around a basic "page" unit, containing information about an individual word, phrase or entity in the world – definitions, thesaurus entries, pronunciation guides and translations in Wiktionary and general biographical, organizational or philosophical information in Wikipedia. In both data sets, pages are linked to each each other and to a user-created category structure – a graph structure where pages are vertices of the graph and page links are the graph edges. We will leverage this graph for determining relatedness.

[1] http://www.wikimedia.org/

Source Word	Alternative Words
make	earn, print, trade, borrow
flawed	imperfect, tiny, lustrous, crude
solitary	alone, alert, restless, fearless

Figure 1: Example TOEFL Questions

Proceedings of the 2009 Workshop on the People's Web Meets NLP, ACL-IJCNLP 2009, pages 28–31,
Suntec, Singapore, 7 August 2009.

2 Extracting Relatedness Measures

We define relatedness based on information flow through the entire Wiktionary graph, rather than by any local in-bound or out-bound link structure. This provides a global measurement of vertex importance, as we do not limit the approach to comparing immediate neighbors.

To do this, we first run the PageRank algorithm (Brin and Page, 1998) iteratively over the graph until convergence to measure the a priori importance of each vertex in graph:

$$\vec{PR}_{t+1} = \alpha \times \left(\vec{PR}_t \cdot E\right) + (1-\alpha) \times \vec{J} \quad (1)$$

In this, E contains the edge transition probabilities, set to a uniform out-bound probability. $\vec{PR}$ holds the PageRank value for each vertex and $\vec{J}$ is uniform vector used to randomly transition between vertices. Traditionally, $\alpha = 0.85$ and is used to tradeoff between a strict transition model and the random-walk model.

We then adopt the extensions proposed in (Ollivier and Senellart, 2007) (**OS**) to determine relatedness given a source vertex:

$$\vec{R}_{t+1} = \alpha \times \left(\vec{R}_t \cdot E + (\vec{S} - \vec{PR})\right) + (1-\alpha) \times \vec{J} \quad (2)$$

$\vec{S}$ is a vector that contains zeros except for a one at our source vertex, and $\vec{PR}$ removes an overall value of 1 based on the a priori PageRank value of the vertex. In this way, vertices close to the source are rewarded with weight and vertices that have a high a priori importance are penalized. When $\vec{R}$ converges, it contains measures of importance for vertices based on the source vertex.

Final relatedness values are then calculated from the vector generated by Equation 2 and the a priori importance of the vector based on the PageRank from Equation 1:

$$rel_{OS}(w, a) = \vec{R}_w[a] \times \log\left(\frac{1}{PR[a]}\right) \quad (3)$$

w is the vertex for the source word and a is the alternative word vertex. The $PR[a]$ penalty is used to further ensure that our alternative vertex is not highly valued simply because it is well-connected.

Applying Equation 3 provides comparable semantic relatedness performance (see Tables 1 and 2). However, cases exist where a single data value is insufficient to make an adequate determination of word relatedness because of small differences for candidate words. We can incorporate additional relatedness information about our vertices by leveraging information about the set of vertices deemed "most related" to our current vertex.

2.1 Integrating N-Best Neighbors

We add information by looking at the similarity between the n-best related words for each vertex. Intuitively, given a source word w and candidate alternatives a_1 and a_2,[2] we look at the set of words that are semantically related to each of the candidates (represented as vectors W, A_1 and A_2). If the overlap between elements of W and A_1 is greater than W and A_2, A_1 is more likely to be the synonym of W.

Highly-ranked shared elements are good indicators of relatedness and should contribute more than low-ranked related words. Lists with many low-ranked words could be an artifact of the data set and should not be ranked higher than ones containing a few high-ranked words.

Our ranked-list comparison metric (**NB**) is a selective mean reciprocal ranking function:

$$rel_{NB}(\vec{W}, \vec{A}, n) = \sum_{r=1}^{n} \frac{1}{r} \times \delta(W_r \in \vec{A}) \quad (4)$$

$\vec{W}$ is the n-best list based on the source vertex and $\vec{A}$ is the n-best list based on the alternative vertex. Values are added to our relatedness metric based on the position of a vertex in the target list and the traditional Dirac δ-function, which has a value of one if the target vertex appears anywhere in our candidate list and a zero in all other cases.

Each metric (OS and NB) will have different ranges. We therefore normalize the reported value by scaling each based on the maximum value for that portion in order to achieve a uniform scale.

Our final metric (**OS+NB**) is created by averaging the two normalized scores. In this work, both scores are given equal weighting. Deriving weightings for combining the two scores will be part of our future work.

$$rel_{OS+NB}(w_{i,j}) = \frac{OS(c_i, c_j) + NB(c_i, c_j, n)}{2} \quad (5)$$

In this, $OS()$ returns the normalized $rel_{OS}()$ value and $NB()$ returns the normalized rel_{NB} value. The maximum $rel_{P+N}()$ value of 1.0 is achieved if c_j has the highest PageRank-based value and the highest N-Best value.

[2] See Figure 1

Source	*ESL* Acc. (%)	*TOEFL* Acc. (%)
JPL	82	78.8
LC-IR	78	81.3
OS	86	88.8
NB	80	88.8
OS+NB	88	93.8

Table 1: ESL and TOEFL Performance

3 Evaluation

We present performance results on three data sets. The first, ESL, uses 50 questions from the English as a Second Language test (Turney, 2001). Next, an 80 question data set from the Test of English as a Foreign Language (TOEFL) is used (Landauer and Dumais, 1997). Finally, we evaluate on the Reader's Digest WordPower (RDWP) data set (Jarmasz and Szpakowicz, 2003). This is a set of 300 synonym detection problems gathered from the Word Power game of the Canadian edition of Reader's Digest Word from 2000 – 2001.

We use the Feb. 03, 2009 version of the English Wiktionary data set [3] for extracting graph structure and relatedness information.

Table 1 presents the performance of our algorithm on the ESL and TOEFL test sets. Our results are compared to Jarmasz and Szpakowicz (2003), which uses a path-based cost on the structure of Roget's Thesaurus (**JPL**) and a cooccurence-based metric, **LC-IR** (Higgins, 2004), which constrained context to only consider adjacent words in structured web queries.

Information about our algorithm's performance on the RDWP test set is found in Table 2. Our results are compared to the previously mentioned algorithms and also the work of Zesch et al. (2008). Their first metric (**ZPL**) uses the path length between two graph vertices for relatedness determination. The second, (**ZCV**), creates concept vectors based on a distribution of pages that contain a particular word.

RDWP is not only larger then the previous two, but also more complictated. TOEFL and ESL average 1.0 and 1.008 number of words in each source and alternative, respectively. For RDWP each entry averages 1.4 words.

We map words and phrases to graph vertices by first matching against the page title. If there is no match, we follow the approach outlined in (Zesch et al., 2008). Common words are removed from the phrase[4] and for every remaining word in the phrase, we determine the page mapping for that individual word. The relatedness of the phrase is then set to be the maximum relatedness value attributed to any of the individual words in the phrase.

Random guessing by an algorithm could increase algorithm performance through random chance. Therefore, we present both a overall percentage and also a precision-based percentage. The first (*Raw*) is defined as the correct number of guesses over all questions. The second (*Prec*) is defined as the correct number of guesses divided by only those questions that were attempted.

3.1 Discussion

For NB and OS+NB, we set $n = 3000$ based on TOEFL data set training.[5] Testing was then performed on the ESL and RDWP data set.

As shown in Table 1, the OS algorithm performs better on the task than the comparison systems. On its own, NB relatedness performs well – at or slightly worse than OS. Combining the two measures increases performance on both data sets. While our TOEFL results are below the reported performance of (Turney et al., 2003) (97.5%), we do not use any task-dependent learning for our results and our algorithms have better performance than any individual module in their system.

Combining OS with NB mitigates the influence of OS when it is not confident. OS correctly picks '*pinnacle*' as a synonym of '*zenith*' with a relatedness value 126,000 times larger than its next competitor. For '*consumed*', OS is wrong, giving '*bred*' a higher score than '*eaten*' – but only by a value 1.2 times that of '*eaten*'. The latter case is overcome by the addition of n-best information while the former is unaffected.

Table 2 demonstrates that we have results comparable to existing state-of-the-art measures. Our choice of n resulted in reduced scores on this task when compared to using the OS metric by itself. But, our algorithm still outperforms both the ZPL and ZCV metrics for our data set in raw scores and in three out of the four precision measures. Further refinement of the RDWP data set mapping or changing our metric score to a weighted sum of

[3] `http://download.wikimedia.org`

[4] Defined here as: {and, or, to, be, the, a, an, of, on, in, for, with, by, into, is, no}

[5] Out of 1.1 million vertices

Metric	Source	Attempted	Score	# Ties	Raw	Prec
JPL	Roget's	300	223	0	.74	.74
LC-IR	Web	300	224.33	-	.75	.75
ZPL	Wikipedia	226	88.33	96	.29	.39
ZCV		288	165.83	2	.55	.58
ZPL	Wiktionary	201	103.7	55	.35	.52
ZCV		174	147.3	3	.49	.85
OS	Wiktionary	300	234	0	.78	.78
NB		300	212	0	.71	.71
OS+NB		300	227	0	.76	.76

Table 2: Reader's Digest WordPower 300 Overall Performance

sorts (rather than a raw maximum) could result in increased performance.

Wiktionary's coverage enables all words in the first two tasks to be found (with the exception of 'bipartisanly'). Enough of the words in the RDWP task are found to enable the algorithm to attempt all synonym detection questions.

4 Conclusion and Future Work

In this paper, we have demonstrated the effectiveness of Wiktionary as a source of relatedness information when coupled with metrics based on information flow using synonym detection as our evaluation testbed.

Our immediate work will be in learning weights for the combination measure, using (Turney et al., 2003) as our guideline. Additional work will be in automatically determining an effective value for n across all data sets.

Long-term work will be in modifying the page transition values to achieve non-uniform transition values. Links are of differing quality, and the transition probabilities should reflect that.

References

Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47.

Derrick Higgins. 2004. Which Statistics Reflect Semantics? Rethinking Synonymy and Word Similarity. In *Proceedings of the International Conference on Linguistic Evidence*.

Mario Jarmasz and Stan Szpakowicz. 2003. Roget's Thesaurus and Semantic Similarity. In *Proceedings of Conference on Recent Advances in Natural Language Processing (RANLP 2003)*.

Thomas K. Landauer and Susan T. Dumais. 1997. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*.

David Milne and Ian H. Witten. 2008. An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *Proceedings of AAAI 2008*.

Yann Ollivier and Pierre Senellart. 2007. Finding Related Pages Using Green Measures: An Illustration with Wikipedia. In *Proceedings of AAAI 2007*.

Michael Strube and Simone Paolo Ponzetto. 2006. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In *AAAI*.

Denis Turdakov and Pavel Velikhov. 2008. Semantic relatedness metric for Wikipedia concepts based on link analysis and its application to word sense disambiguation. In *Proceedings of CEUR*.

Peter D. Turney, Michael L. Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining Independent Modules in Lexical Multiple-Choice Problems. In *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*.

Peter D. Turney. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, pages 491–502, Freidburg, Germany.

Torsten Zesch, Christof Muller, and Iryna Gurevych. 2008. Using Wiktionary for Computing Semantic Relatedness. In *Proceedings of AAAI 2008*.