Duluth at SemEval-2017 Task 7: Puns Upon a Midnight Dreary, Lexical Semantics for the Weak and Weary

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

Abstract

This paper describes the Duluth systems that participated in SemEval-2017 Task 7 : Detection and Interpretation of English Puns. The Duluth systems participated in all three subtasks, and relied on methods that included word sense disambiguation and measures of semantic relatedness.

1 Introduction

Puns represent a broad class of humorous word play. This paper focuses on two types of puns, *homographic* and *heterographic*.

A *homographic* pun is characterized by an oscillation between two senses of a single word, each of which leads to a different but valid interpretation:

> I'd like to tell you a chemistry joke but I'm afraid of your reaction.

Here the oscillation is between two senses of *reaction*. The first that comes to mind is perhaps that of a person revealing their true feelings about something (how they react), but then the relationship to *chemistry* emerges and the reader realizes that *reaction* can also mean the chemical sense, where substances change into others.

Homographic puns can also be created via compounding:

> He had a collection of candy that was in mint condition.

The pun relies on the oscillation between the flavor *mint* and the compound *mint condition*, where *candy* interacts with *mint* and *mint condition* interacts with *collection*.

A *heterographic* pun relies on a different kind of oscillation, that is between two words that nearly sound alike, rhyme, or are nearly spelled the same.

> The best angle from which to solve a problem is the try angle.

Here the oscillation is between *try angle* and *triangle*, where *try* suggests that the best way to solve a problem is to try harder, and *triangle* is (perhaps) the best kind of angle.

This example illustrates one of the main challenges of heterographic puns, and that is identifying multi word expressions that are used as a kind of compound, but without being a standard or typical compound (like the very non-standard *try angle*). One reading treats *try angle* as a kind of misspelled version of *triangle* while the other treats them as two distinct words (*try* and *angle*). There is also a kind of oscillation between senses here, since *try angle* can waver back and forth between the geometric sense and the one of making effort.

During our informal study of both heterographic and homographic puns, we observed a fairly clear pattern where a punned word will occur towards the end of a sentence and has a sense that is semantically related to an earlier word, and another sense that fits the immediate context in which it occurs. It often seemed that the sense that fits the immediate context is a more conventional usage (as in *afraid of your reaction*) and the more amusing sense is that which connects to an earlier word via some type of semantic relation (*chemical reaction*). This is more complicated in the case of heterographic puns since the punned word can rely on pronunciation or spelling to create the effect (i.e., *try angle* versus *triangle*). In this work we focused on exploiting these long distance semantic relations, although in future work we plan to consider the use of language models to identify more conventional usages.

We used two versions of the WordNet SenseRelate word sense disambiguation algorithm[1] : Tar-

[1] http://senserelate.sourceforge.net

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 416–420,
Vancouver, Canada, August 3 - 4, 2017.

getWord (Patwardhan et al., 2005) and AllWords (Pedersen and Kolhatkar, 2009). Both have the goal of finding the assignment of senses in a context that maximizes their overall semantic relatedness (Patwardhan et al., 2003) according to measures in WordNet::Similarity[2] (Pedersen et al., 2004). We relied on the Extended Gloss Overlaps measure (lesk) (Banerjee and Pedersen, 2003) and the Gloss vector measure (vector) (Patwardhan and Pedersen, 2006).

The intuition behind a Lesk measure is that related words will be defined using some of the same words, and that recognizing these overlaps can serve as a means of identifying relationships between words (Lesk, 1986). The Extended Gloss overlap measure (hereafter simply lesk) extends this idea by considering not only the definitions of the words themselves, but also concatenates the definitions of words that are directly related via hypernym, hyponym, and other relations according to WordNet.

The Gloss Vector measure (hereafter simply vector) extends this idea by representing each word in a concatenated definition with a vector of co-occurring words, and then creating a representation of this definition by averaging together all of these vectors. The relatedness between two word senses can then be measured by finding the cosine between their respective vectors.

2 Systems

The evaluation data for each subtask was individual sentences that are independent of each other. All sentences were tokenized so that each alphanumeric string was separated from any adjacent punctuation, and all text was converted to lowercase. Multi-word expressions (compounds) found in WordNet were identified.

SemEval–2017 Task 7 (Miller et al., 2017) focused on pun identification, and was divided into three subtasks.

2.1 Subtask 1

The problem in Subtask 1 was to identify if a sentence contains a pun (or not). We relied on the premise that a sentence will have one unambiguous assignment of senses, and that this should be true even as the parameters of a word sense disambiguation algorithm are varied. Thus, if a sentence has multiple possible assignments of senses based on the results of different runs of a word sense disambiguation algorithm, then there is a possibility that a pun exists. To investigate this hypothesis we ran the WordNet::SenseRelate::AllWords algorithm using four different configurations, and then compared the four sense tagged sentences with each other. If there were more than two differences in the sense assignments that resulted from these different runs, then the sentence is presumed to contain a pun.

WordNet::SenseRelate::AllWords takes measures of semantic relatedness between all the pairwise combinations of words in a sentence that occur within a certain number of positions of each other (the window size), and assigns the sense to each content word that results in the maximum relatedness among the words in that window. The assumption that underlies this method is that words in a window will be semantically related, at least to an extent, so when choices among word senses are made, those that are most related to other words in the window will be selected.

The four configurations include two where the window of context is the entire sentence (a wide window) and another two where the window of context is only one word to the left and one word to the right (a narrow window). In addition these two configurations were carried out with and without compounding of words being performed prior to disambiguation. In all four configurations the Gloss Vector measure WordNet::Similarity::vector was used as the measure of semantic relatedness. If more than two sense changes result from these different configurations, then we say that a pun has occurred in the sentence.

2.2 Subtask 2

In Subtask 2 the evaluation data consists of the instances from Subtask 1 that contain puns. The task is to identify the punning word.

We took two approaches to this subtask, however both were informed by our observation that punned words often occur later in sentences. The first (run 1) was to rely on our word sense disambiguation results from Subtask 1 and identify the last word which changed senses between different runs of the WordNet::SenseRelate::AllWords disambiguation algorithm. We relied on two of the four configurations used in Subtask 1. We used the narrow and wide contexts from Subtask 1 without

[2] http://wn-similarity.sourceforge.net

finding compounds. We realized that this might cause us to miss some cases where a pun was created with a compound, but our intuition was that the more common cases (especially for homographic puns) would be those without compounds. Our second approach (run 2) was a simple baseline where the last content word in the sentence was simply assumed to be the punned word.

2.3 Subtask 3

The evaluation data for Subtask 3 includes heterographic and homographic instances from Subtask 2 where the word being punned has been identified. The task is to determine which two senses of the punned word are creating the pun.

We used the word sense disambiguation algorithm WordNet::SenseRelate::TargetWord, which assigns a sense to a single word in context (whereas AllWords assigns a sense to every word in a context). However, both TargetWord and AllWords have the same underlying premise, and that is that words in a sentence should be assigned the senses that are most related to the senses of other words in that sentence.

We tried various combinations of TargetWord configurations, where each would produce their own verdict on the sense of the punned word. We took the two most frequent senses assigned by these variations and used them as the sense of the punned word. Note that for the heterographic puns there was an additional step, where alternative spellings of the target word were included in the disambiguation algorithm. For example :

The dentist had a bad day at the orifice.

Orifice is already identified as the punned word, and one of the intended senses would be that of an opening, but the other is the somewhat less obvious spelling variation *office*, as in *a bad day at the office*.

For the first variation (run 1) we used both the local and global options from TargetWord. The local option measures the semantic relatedness of the target word with all of the other members of the window of context, whereas the global option measures the relatedness among all of the words in the window of context (not just the target word). We also varied whether the lesk or vector measure was used, if a narrow or wide window was used, and if compounds were identified. We took all possible combinations of these variations, which resulted in 16 possible configurations. To this we added a WordNet sense one baseline with and without finding compounds, and a randomly assigned sense baseline. Thus, there were 19 variations in our run 1 ensemble. We took this approach with both the homographic and heterographic puns, although for the heterographic puns we also replaced the target word with all of the words known to WordNet that differed by one edit distance. The premise of this was to detect minor misspellings that might enable a heterographic pun.

For run 2 we only used the local window of context with WordNet::SenseRelate::TargetWord, but added to lesk and vector the Resnik measure (res) and the shortest path (path) measure. We carried out each of these with and without identifying compounds, which gives us a total of eight different combinations. We also tried a much more ambitious substitution method for the heterographic puns, where we queried the Datamuse API in order to find words that were rhymes, near rhymes, homonyms, spelled like, sound like, related, and means like words for the target word. This created a large set of candidate target words, and all of these were disambiguated to find out which sense of which target word was most related to the surrounding context.

3 Results

We review our results in the three subtasks in this section. Table 1 refers to homographic results as *hom* and heterographic as *het*. Thus the first run of the Duluth systems on homographic data is denoted as Duluth-hom1, and the first run on heterographic data is Duluth-het1. The highest ranking system is indicated via High-hom and High-het. P and R as column headers stand for precision and recall, A stands for accuracy, and C is for coverage. Rank x/y indicates that this system was ranked x of y participating systems.

3.1 Subtask 1

Puns were found in 71% (1,271) of the heterographic and 71% of the homographic instances (1,607). This suggests this subtask would have a relatively high baseline performance, for example if a system simply predicted that every sentence contained a pun. Given this we do not want to make too strong a claim about our approach, but it does seem that focusing on sentences that have multiple possible (and valid) sense assignments

Table 1: Subtask 1, 2, 3 results

Subtask 1	P	R	A	F1	rank
High-hom	.97	.80	.84	.87	1 / 9
Duluth-hom1	.87	.78	.74	.83	2 / 9
High-het	.87	.82	.78	.84	1 / 7
Duluth-het1	.87	.74	.69	.80	3 / 7
Subtask 2	P	R	C	F1	rank
High-hom	.66	.66	1.0	.66	1 / 15
Duluth-hom1	.37	.36	.99	.37	7 / 15
Duluth-hom2	.44	.44	1.0	.44	6 / 15
High-het	.80	.80	1.0	.80	1 / 11
Duluth-het1	.18	.18	.99	.18	11 / 11
Duluth-het2	.53	.53	1.0	.53	4 / 11
Subtask 3	P	R	C	F1	rank
High-hom	.17	.14	.86	.16	1 / 8
Duluth-hom2	.17	.14	.86	.16	1 / 8
Duluth-hom1	.15	.15	1.0	.15	3 / 8
High-het	.08	.07	.83	.08	1 / 6
Duluth-het1	.03	.03	1.0	.03	3 / 6
Duluth-het2	.001	.001	.98	.001	6 / 6

is promising for pun identification. Our method tended to over-predict puns, reporting that a pun occurred in 84% (1,489 of 1,780 instances) of the heterographic data, and 80% (1,791 of 2,250 instances) of the homographic.

3.2 Subtask 2

Subtask 2 consists of all the instances from Subtask 1 that included a pun. This leads to 1,489 heterographic puns and 1,791 homographic.

We see that our simple baseline method of choosing the last content word as the punned word (run 2) significantly outperformed our more elaborate method (run 1) of identifying which word experienced more changes of senses across multiple variations of the disambiguation algorithm. We can also see that run 1 did not fare very well with heterographic puns. In general we believe the difficulty that run 1 experienced was due to the overall noisiness that is characteristic of word sense disambiguation algorithms.

3.3 Subtask 3

Subtask 3 consists of 1,298 homograph instances and 1,098 heterographic instances. We see that for homographs our method fared very well, and was the top ranked of participating systems. On the other hand our heterographic approach was not terribly successful. We believe that the idea of generating alternative target words for heterographic puns is necessary, since without this it would be impossible to identify one of the senses of the punned word. However, our run 1 approach of simply using target word variations with an edit distance of one did not capture the variations present in heterographic puns (e.g., *orifice* and *office* have an edit distance of 2). Our run 2 approach of finding many different target words via the Datamuse API resulted in an overwhelming number of possibilities where the intended target word was very difficult to identify.

4 Discussion and Future Work

One limitation of our approach is the uncertain level of accuracy of word sense disambiguation algorithms, which vary from word to word and domain to domain. Finding multiple possible senses for a single word may signal a pun or expose the limits of a particular WSD algorithm.

In addition, the contexts used in this evaluation were all single sentences, and were relatively short. Whether or not having more context available would help or hinder these approaches is an interesting question.

Heterographic puns posed a host of challenges, in particular mapping clever near spellings and near pronunciations into the intended form (e.g., *try angle* as *triangle*). Simply trying to assign senses to *try angle* will obviously miss the pun, and so the ability to map similar sounding phrases to the intended word is a capability that our systems were not terribly successful with. However, we were better able to identify compounds in homographic puns (e.g., *mint condition*) since those were written literally and could be found (if in WordNet) via a simple subsequence search.

While our reliance on word sense disambiguation and semantic relatedness served us well for homographic puns, it was clearly not sufficient for heterographic. Moving forward it seems important to have a reliable mechanism to map the spelling and pronunciation variations that characterize heterographic puns to their intended forms. While dictionaries of rhyming and sound-alike words are certainly helpful, they typically introduce too many possibilities from which to make a reliable selection. Language modeling seems like a promising way to winnow that space, so that we can get from a *try angle* to a *triangle*.

References

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*. Acapulco, pages 805–810.

M.E. 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*. ACM Press, pages 24–26.

Tristan Miller, Christian F. Hempelmann, and Iryna Gurevych. 2017. SemEval-2017 Task 7: Detection and interpretation of English puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, BC.

S. Patwardhan, S. Banerjee, and T. 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*. Mexico City, pages 241–257.

S. Patwardhan, S. Banerjee, and T Pedersen. 2005. SenseRelate::TargetWord - a generalized framework for word sense disambiguation. In *Proceedings of the Demonstration and Interactive Poster Session of the 43rd Annual Meeting of the Association for Computational Linguistics*. Ann Arbor, MI, pages 73–76.

S. Patwardhan and T. 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*. Trento, Italy, pages 1–8.

T. Pedersen and V. Kolhatkar. 2009. WordNet::SenseRelate::AllWords - a broad coverage word sense tagger that maximizes semantic relatedness. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies 2009 Conference*. Boulder, CO, pages 17–20.

T. Pedersen, S. Patwardhan, and J. 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*. Boston, MA, pages 38–41.