PAE-DIRT: Paraphrase Acquisition Enhancing DIRT

Michael Chung and David Hall
Stanford University
{mcfenix,dlwh}@stanford.edu

Abstract

We present several algorithms for detecting sentence similarity in clustered news articles, hopefully ones that Dekang Lin's DIRT system can use. These include simple ngram overlap, unigram-idf overlap, LSA-based cosine similarity overlap, and dependency-parse link overlap. We also present a comparison of the algorthms against a gold standard hand-chosen similar sentences.

Introduction

Dekang Lin and Patrick Pantel's DIRT (2001) algorithm discovers inference rules from text by storing dependency triples from training sentences and using them as features to calculate dependency path similarities. The triples are of the form (path, slot, word), where slot is one of the two ends of the path and word is the filler of the slot. The idea behind DIRT is the Extended Distributional Hypothesis, which states that dependency paths that link to the same sets of words have similar meanings. So for example:

The man walked into the store.
The man went inside the store.

DIRT would notice that both dependency paths above link to the words, “man” and “store”, and would therefore consider “X walked into Y” and “X went inside Y” to be more similar. This is useful for generating inference rules, but unfortunately, DIRT is not very accurate. One big reason for this inaccuracy is the fact that there are plenty of dependency paths that share similar

sets of slot fillers but are not alike in meaning. Here is an example:

The man walked into the store.
The man barged into the store.

And another...

The man asked the woman.
The man informed the woman.

It is apparent that finding dependency path similarities using slot fillers as features is not good enough to properly determine whether two paths share similar meanings. With PAE-DIRT, we aim to amend this weakness of DIRT by detecting paraphrases, so that DIRT may distinguish "walked" from "barged".

Previous Work

MUD: Making Useful DIRT (Steinberg, 2005) is a modification of the DIRT algorithm that assigns scores to sentence pairs (rather than generating rules given a sentence) and handles polarity issues. This is similar to our goals, with a few differences. Instead of modifying the DIRT algorithm, we are trying to detect paraphrases to improve the performance of DIRT. Also, while MUD specifically aims to fix DIRT's tendency to pair up dependency paths with opposite meanings, we try to tackle the more general problem of distinguishing dependency paths, whether they are opposite or simply different in meaning.

Others have looked at using news corpora for similarity or entailment pairs. For example, Burger and Ferro (2005) extracted pairs from headlines and the lead paragraphs of articles for training their RTE system. Shinyama, et al. (2002) explored parallel news corpora, but limited their extractions to domains heavy with named entities (arrests and personnel affairs.) Our project differs from the former in that we wish to extract from documents that actually contain

paraphrases of one another, and from the latter in that we want a broad area measure of sentence similarity that does not depend on named entities to function.

Methods

We examined a number of different algorithms:

Unigram-IDF Overlap, Stopped and Unstopped.

Our first metric is adapted from Shinyama, et al. (2002) for paraphrase acquisition for information extraction. In their paper, they described a similarity score:

$$score_S(s_{1,}s_2)=\cos(W_{1,}W_2)$$
$$W^i=TF(w_i)*IDF(w_i)$$
$$idf(w)=\log\left(\frac{|Documents|}{|\{d\in Documents: w\in d\}|}\right)$$

For their paper, they focused in the categories of arrests and personnel affairs, in which almost all nouns involve named entities. However, we pursued a more generalized notion of similarity, so we chose to use all words. Thus, our metric calculates the sum of idf scores of word shared by the two sentences divided by the idf scores of the shared words.

$$unigram-alignment(s_{1,}s_2)=\frac{\sum_{w\in s_1\cap s_2} idf(w)}{\sum_{w\in s_1\cup s_2} idf(w)}$$

This metric—like most of our metrics—has the useful property of returning a number in the range [0,1], returning 1 when $s_1=s_2$. Our implementation employed a stop list so that many sentences which differed only slightly (e.g. the presence of a single determiner) were actually found to be identical. For DIRT, we can then help to eliminate sentences that cannot be used for the extraction of inference rules (by filtering out sentences with similarity 1.)

N-Gram Overlap

Our next metric was inspired by the BLEU score from Papenini, et al., 2002. Usually, the BLEU score is employed in machine translation to gauge how much overlap a candidate translation has with a gold-standard translation. It is defined as

$$overlap-score_N(s_1, s_2) = b_N(s_1, s_2) \exp \sum_{i=1}^{N} w_i \log precision_i(s_1, s_2)$$

with w_i defined as either $1/N$ or $1/2^{i-1}$ and b_N as a brevity penalty. (We used 1/N.) However, in considering paraphrases, precision is ill defined. Moreover, sentences that contain paraphrases of another sentence should not be more heavily penalized than necessary. Thus, we did away with any brevity penalty. Thus, we developed this metric:

$$score_n(s_1, s_2) = \frac{|\{w_i \ldots w_{i+n-1} \in s_1 \cap s_2\}|}{|\{w_i \ldots w_{i+n-1} \in s_1 \cup s_2\}|}$$

$$overlap-score_N(s_1, s_2) = \exp \sum_{i=1}^{N} w_i \log score_i(s_1, s_2)$$

In our implementation, we actually employed our unigram-idf score for n=1, and we tested versions with N=2, N=3, and N=4.

LSA Cosine Similarity

Our next metric uses the Infomap package trained on a large fraction of the Wikipedia to generate cosine scores. The metric itself is:

$$half-score(s_f, s_t) = \sum_{w_f \in s_f} \max_{w_t \in s_t} infomap-cosine(w_f, w_t)$$

$$infomap-score(s_1, s_2) = half-score(s_1, s_2) \cdot half-score(s_1, s_2)$$

This metric takes advantage of the notion that paraphrases really do not need the same words at all—though that's of course preferable—and thus it attempts to maintain some level of similarity through cosine distance.

LSA Cosine Similarity-IDFs

We also augmented the formula above to use idf scores to weight more important words. This gives the same weighting to very rare words (for instance, proper names) while providing the benefits of lining up similar more common words. The change is rather small:

$$half-score(s_f, s_t) = \frac{\sum_{w_f \in s_f} idf(w_f) \cdot \max_{w_t \in s_t} infomap-cosine(w_f, w_t)}{\sum_{w_f \in s_f} idf(w_f)}$$

Dependency Link Overlap

We designed an algorithm that scores sentence similarity based on the percentage of shared dependency links. We define a dependency link to be a triple consisting of two adjacent words in a sentence and the dependency path (of length one) between them. Using Dekang's Lin Minipar to extract a set of dependency links for each sentence, the algorithm counts the number of links shared by two sentences and divides it by the number of links in the greater set of the two.

$$score_{max}(S_1, S_2) = \frac{|S_1 \cap S_2|}{max(|S_1|, |S_2|)}$$

We also try dividing by the size of the smaller set instead, to capture paraphrases where a small sentence is very similar to a part of a much longer sentence.

$$score_{min}(S_1, S_2) = \frac{|S_1 \cap S_2|}{min(|S_1|, |S_2|)}$$

While this helps to claim certain paraphrases that would otherwise be unrecognized, it can also result in erroneous high score assignments to pairings in which short sentences share most of their dependency links with long sentences, but the longer sentences do not have much in common with the shorter sentences.

These metrics can also have trouble when confronted with sentences that look similar but are not paraphrases of each other due to having different verbs. This is especially an issue when the algorithm is used with the Wikipedia corpus. We tried penalizing sentence pairs with different verbs by multiplying their scores by 0.7, at the risk of losing correct paraphrases that do not share verbs.

$$score_{maxpenalty}(S_{1,}S_2)=\frac{|S_1\cap S_2|}{max(|S_1|,|S_2|)}*\begin{Bmatrix}0.7\,if\ verbs\,differ\\ 1.0\,otherwise\end{Bmatrix}$$

Experimental Setup

For testing, we opted not to follow the Dekang Lin's measures of effectiveness, instead opting to analyze only our system's ability to find similar sentences. the authors hand-aligned four articles in a cluster independently of one another, calculated kappa score, and evaluated our various systems against the annotations by using a thresholding system, which gave us a simple binary similar/not similar answer. Thus, we could establish precision/recall curves.

Corpus Selection

We chose for our initial corpus the Wikipedia based on the assumption that an encyclopedia would contain several summarization sentences. However, that assumption turned out to be mostly untrue: longer, biographical articles tended to contain such sentences, but the vast majority of articles (based on informal analysis) did not contain any summarization at all.

Instead, we focused on hand-created news clusters from SummBank 1.0 and Cross-Document Structure Theory Bank, totaling 97 clusters, each with approximately 10 documents.

Gold-Standard Creation

We held out one cluster and hand chose paraphrases from four of the documents independently from one another. We marked alignments as “sure” and “possible”, evaluated

kappa, and then compared answers to come up with a true gold standard annotation. We determined a kappa of 0.98, which indicates very strong inter-annotator agreement. However, we feel that our calculation may be a bit high, since many sentences are not feasibly alignable. A more reasonable measure would limit the domain of sentences that could be aligned.

Results

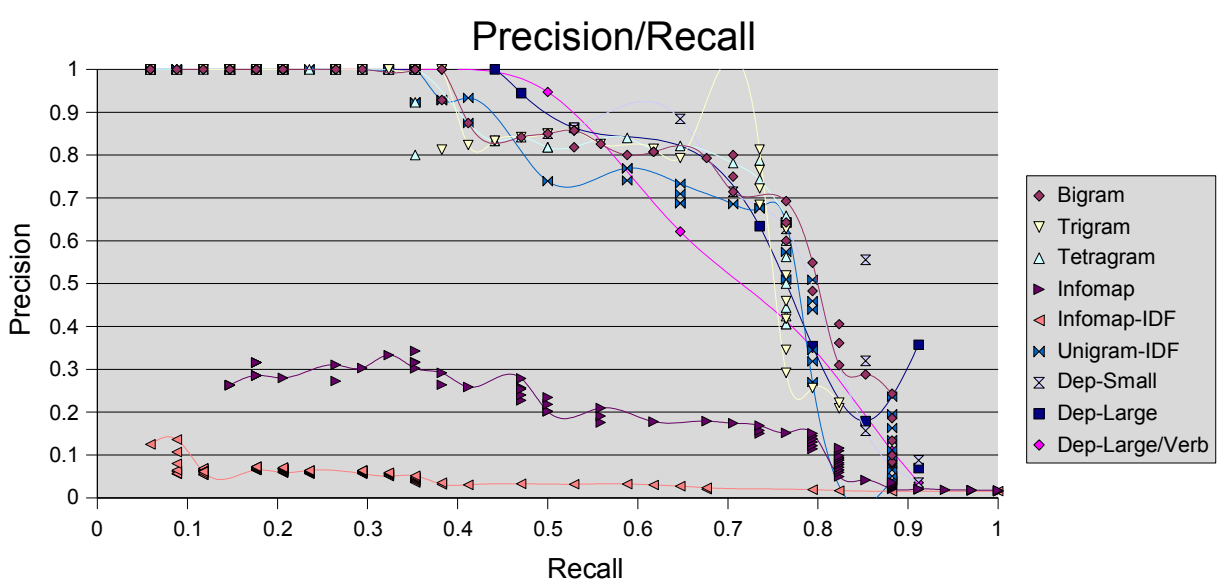
We start by presenting our results in a precision/recall curve. We present one with individual data points, and one without data points and without the infomap scores, which were considerably lower. We define precision and recall as

$$Precision = \frac{|P \cap A| + |S \cap A|}{|A|}$$

$$Recall = \frac{|S \cap A|}{|S|}$$

We also calculate F1-Scores, which is:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$



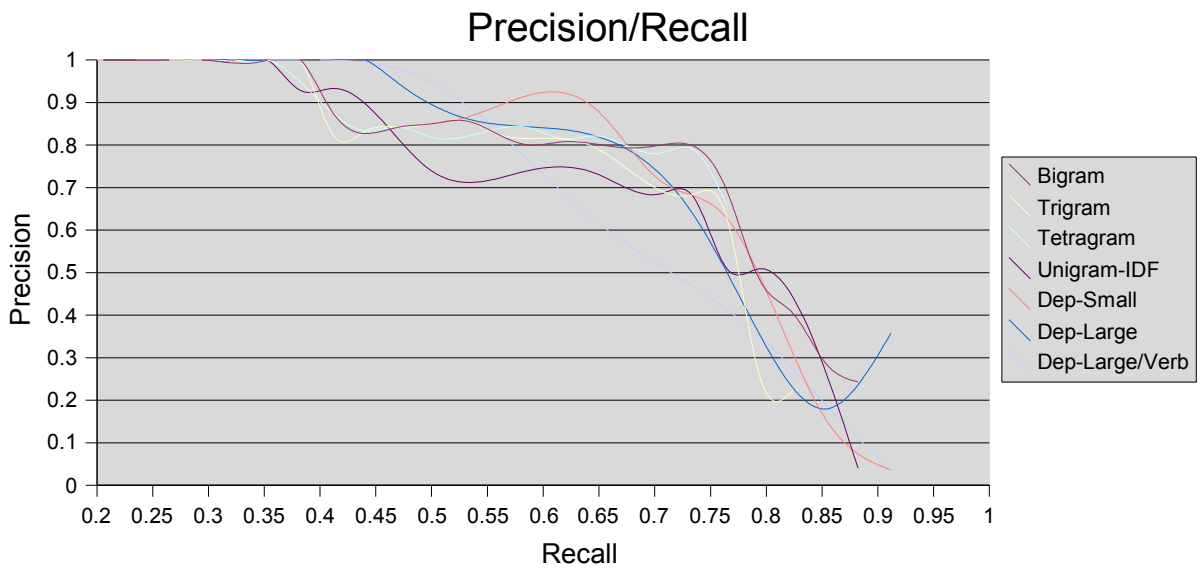


Table 1: Best Scores

	Threshold	***Precision***	***Recall***	***F-Measure***
Unigram-IDF	0.24	0.68	0.74	0.7
Bigram	0.20	0.71	0.80	0.75
Trigram	0.16	0.81	0.74	0.77
Tetragram	0.12	0.79	0.74	0.76
Infomap	0.76	0.34	0.35	0.35
Infomap-IDF	0.69	0.28	0.47	0.35
Dependency-penalize	0.19	0.95	0.53	0.68
Dependency-bigger	0.19	0.63	0.76	0.69
Dependency-smaller	0.40	0.88	0.65	0.75

Analysis

On this test set, the trigram metric turned out to have the best F-Measure, though the dependency-smaller and the tetragram were not significantly different. Interestingly, the ngram metrics seem to be more balanced than the syntax-based metrics, with precision and recall very close in value.

Problems with the ngram overlap metrics come exactly where one would expect: phrases that are quite similar except for the addition or deletion of a few insignificant words. For example, consider the following pair of almost identical sentences:

> The winners will be awarded trophies cameras and photo developing cash coupons.
>
> Winners will be awarded trophies cameras and photo developing cash coupons.

These have a tetragram alignment score of 0.77, while simply removing the initial “the” would raise the similarity score to 1.0. In particular, such small differences could slow down a system like DIRT with extraneous sentences or unfairly weight certain sentences that are practically identical. An initial check to remove sentences that are approximately substrings of one another could help.

As expected, penalizing sentence pairs with different verbs had the effect of increasing accuracy but lowering the recall. Without the penalty, the algorithm is less picky and finds more correct paraphrases, in exchange for reduced precision. The Dependency-Smaller measure is not as precise as the other (non-LSA-based) metrics – it never reaches 100% precision. As mentioned before, this is due to the fact that an unreasonably high score is assigned when a high percentage of a short sentence's dependency links is shared with a low percentage of a longer sentence's links. Consider these examples of sentences aligned by the metric at a threshold of 0.50 :

Ex. 1

> The Government is firmly committed to promoting the development of electronic commerce in Hong Kong, especially amongst the small and medium sized enterprises.
>
> I will talk about the development of electronic commerce in Hong Kong.

Ex. 2

> He was confident that the Government initiatives he outlined would create a favourable environment for the development of electronic commerce in Hong Kong.
>
> I will talk about the development of electronic commerce in Hong Kong.

In both cases, the phrase, "the development of electronic commerce in Hong Kong", is shared between two sentences. The metric assigns a high score to both examples because the phrase is a significant portion of the smaller sentence. However, these paraphrases are obviously incorrect. The verb differences are significant enough in these sentence pairs that the meanings of the sentences are dissimilar, despite the similarities of the other words besides the verbs. A penalty for verb difference would solve these errors, but a very severe reduction would be required to counteract the high score determined by the algorithm, and such a severe penalty applied uniformly on all sentence pairs does not result in any improvement overall. One idea would be to vary the penalty depending on a measure of similarity between different verbs, such as LSA or VerbNet.

The generalized Infomap metrics faired surprisingly poorly, mostly from a bad showing in precision. (Poor precision shows up not just in the evaluation of precision, but in the high threshold score.) Reconsideration of the metric revealed that this is not entirely unexpected, since LSA allows more words to overlap than any of the other metrics. We tried weighting similarities (by raising the cosine to various powers) without any real success. Another possibility would be to threshold Infomap scores so that words that were below a certain degree of similarity would

be ignored. This could be further augmented by training Infomap on tagged corpora, and then tagging the input clusters. Finally, the Infomap metric could be restricted in its application, only being applied to adjectives, adverbs or verbs (the things that varied the most from document to document in a cluster) while things that are noun-like would not be tested for cosine similarity.

Conclusion/Future Research

From a purely numerical perspective, the ngram overlap metrics seem to have the best performance, and require the least amount of code to create the calculations. However, the dependency overlap metric is conceivably quite useful since the DIRT system must generate these anyway. Another possibility would be to take a first pass with a high recall/high speed metric, and then calculate again with a dependency metric.

Thus, the most obvious direction for future research is to try the various metrics with DIRT itself. In addition to simply choosing all paraphrases above a certain threshold, one could employ the scores to create partial counts for DIRT's triple database. As we stated before, evaluation of paraphrases is in general a fuzzy concept—at least in the manner presented in Lin (2001). Thus, a more careful evaluation method would need to be defined before exploring performance in DIRT precisely. Other directions could involve baselines for various other tasks, such as RTE or IE. While we are left with many options for future investigation, in this work we have shown that unsupervised methods can be used to successfully detect paraphrases.

References

Burger, John and Lisa Ferro, 2005. Generating an Entailment Corpus from News Headlines. ACL Proceedings on Empirical Modeling of Semantic Equivalence and Entailment.

Lin, Dekang and Patrick Pantel. 2001. DIRT-Discovery of Inference Rules from Text. ACM SIGKDD.

Papineni, Kishore, et al. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting on ACL.

Radev, Dragomir, et al. 2003. SummBank 1.0. Linguistic Data Consortium, Philadelphia.

Radev, Dragomir, et al. 2003. CSTBank: Cross-document Structure Theory Bank.

Shinyama, Yusuke, et al. 2002. Automatic Paraphrase Acquisition from News Articles. HLT.

Steinberg, Ari. 2005. MUD: Making Useful Dirt.