DIT: Summarisation and Semantic Expansion in Evaluating Semantic Similarity

Magdalena Kacmajor
IBM Technology Campus
Dublin Software Lab
Ireland
magdalena.kacmajor@ie.ibm.com

John D. Kelleher
Applied Intelligence Research Centre
Dublin Institute of Technology
Ireland
john.d.kelleher@dit.ie

Abstract

This paper describes an approach to implementing a tool for evaluating semantic similarity. We investigated the potential benefits of (1) using text summarisation to narrow down the comparison to the most important concepts in both texts, and (2) leveraging WordNet information to increase usefulness of cosine comparisons of short texts. In our experiments, text summarisation using a graph-based algorithm did not prove to be helpful. Semantic and lexical expansion based upon word relationships defined in WordNet increased the agreement of cosine similarity values with human similarity judgements.

1 Introduction

This paper describes a system that addresses the problem of assessing semantic similarity between two different-sized texts. The system has been applied to SemEval-2014 Task 3, Cross-Level Semantic Similarity (Jurgens et al, 2014). The application is limited to a single comparison type, that is, paragraph to sentence.

The general approach taken can be characterised as text summarisation followed by a process of semantic expansion and finally similarity computation using cosine similarity.

The rationale for applying summarisation is to focus the comparison on the most important elements of the text by selecting key words to be used in the similarity comparison. This summarisation approach is based on the assumption that if summary of a paragraph is similar to the summary sentence paired with the paragraph in the task dataset, then the original paragraph and sentence pair must have been similar and so should receive a high similarity rating.

The subsequent semantic expansion is intended to counteract the problem arising from the small size of both compared text units. The similarity metric used by the system is essentially a function of word overlap. However, because both the paragraphs and sentences being compared are relatively short, the probability of a word overlap - even between semantically similar texts - is quite small. Therefore prior to estimating the similarity between the texts we extend the word vectors created by the summarisation process with the synonyms and other words semantically and lexically related to the words occurring in the text.

By using cosine similarity measure, we normalize the lengths of word vectors representing different-sized documents (paragraphs and sentences).

The rest of the paper is organized as follows: section 2 describes the components of the system in more detail; section 3 describes parameters used in the experiments we conducted and presents our results; and section 4 provides concluding remarks.

2 System Description

2.1 Overview

There are four main stages in the system processing pipeline: (1) text pre-processing; (2) summarisation; (3) semantic expansion; (4) computing the similarity scores. In the following sections we describe each of these stages in turn.

2.2 Pre-processing

Paraphrases and sentences are tokenized and annotated using Stanford CoreNLP\textsuperscript{1}. The annotations include part-of-speech (POS), lemmatisation, dependency parse and coreference resolution. Then

\textsuperscript{1}http://nlp.stanford.edu/software/corenlp.shtml
the following processes are applied: (a) **Token selection**, the system ignores tokens with POS other than nouns, adjectives and verbs (in our experiments we tested various combinations of these three categories); (b) **Token merging**, the criteria for merging can be more restrictive (same word form) or less restrictive – based on same lemma, or even same lemma ignoring POS; (c) **Stopword removal**, we apply a customized stopword list to exclude verbs that have very general meaning. In this way, each text unit is processed to produce a filtered set of tokens which at the next step can be directly transformed into nodes in the graph representation of the text. Dependency and coreference annotations will be used for defining edges in the graph.

### 2.3 Summarisation system

Summarisation has been implemented using TextRank (Mihalcea and Tarau, 2004), an iterative graph-based ranking algorithm derived from PageRank (Brin and Page, 1998).

The ranking is based on the following principle: when node \( i \) links to node \( j \), a vote is cast that increases the rank of node \( j \). The strength of the vote depends on the importance (rank) of the casting node, thus the algorithm is run iteratively until the ranks stop changing beyond a given threshold, or until a specified limit of iterations is reached.

To apply this algorithm to paragraphs and sentences, our system builds a graph representation for each of these text units, with nodes representing the tokens selected and merged at the preceding stage. The nodes are connected by co-occurrence (Mihalcea and Tarau, 2004), dependency and/or coreference relations. Next, a unweighted or weighted version of the ranking algorithm is iterated until convergence.

For each test unit, the output of the summariser is a list of words sorted by rank. Depending on the experimental setup, the summariser forwards on all processed words, or only a subset of top-ranked words.

### 2.4 Lexico-semantic expansion

For each word returned from the summariser, we retrieve all (or a predetermined number of) synsets that have this word as a member. For each retrieved synset, we also identify synsets related through semantic and lexical relations. Finally, using all these synsets we create the synonym group for a word that includes all the members of these synsets.

If a word has many different senses, then the synonym group grows large, and the chances that the sense of a given member of this large group will match the sense of the original word are shrinking. To account for this fact, each member of the synonym group is assigned a weight using Equation 1. This weight is simply 1 divided by the count of the number of words in the synonym group.

\[
synweight = \frac{1}{\#\text{SynonymGroup}} \quad (1)
\]

At the end of this process for each document we have the set of words that occurred in the document, and each of these words has a synonym group associated with it. All of the members of the synonym groups have a weight value.

### 2.5 Similarity comparison

Cosine similarity is used to compute the similarity for each paragraph-sentence pair. For this calculation each text (paragraph or sentence) is represented by a bag-of-words vector containing all the words derived from the text together with their synonym groups.

The bag-of-words can be binary or frequency based, with the counts optionally modified by the word ranks. The counts for words retrieved from WordNet are weighted with \( \text{synweights} \), which means that they are usually represented by very small numbers. However, if a match is found between a WordNet word and a word observed in the document, the weight of both is adjusted according to semantic match rules. These rules have been established empirically, and are presented in section 3.3.1.

The cosine values for each paragraph-sentence pair are not subject to any further processing.

### 3 Experiments

#### 3.1 Dataset

All experiments were carried out on training dataset provided by SemEval-2014 Task 3 for paragraph to sentence comparisons.

#### 3.2 Parameters

For each stage in the pipeline, there is a set of parameters whose values influence the final results. Each set of parameters will be discussed next.
3.2.1 Pre-processing parameters
The parameters used for pre-processing determine
the type and number of nodes included in the
graph:

- **POS**: Parts-of-speech that are allowed into
  the graph, e.g. only nouns and verbs, or
  nouns, verbs and adjectives.

- **Merging criteria**: The principle by which
  we decide whether two tokens should be rep-
  resented by the same node in the graph.

- **Excluded verbs**: The contents of the stop-
  word list.

3.2.2 Summarisation parameters
These parameters control the structure of the graph
and the results yielded by TextRank algorithm.
The types of nodes in the graph are already de-

cided at the pre-processing stage.

- **Relation type**: In order to link the nodes
  (words) in the graph representation of a docu-
  ment, we use co-occurrence relations (Mihal-
  cea and Tarau, 2004), dependency relations
  and coreference relations. The two latter are
  defined based on the Stanford CoreNLP an-
  notations, whereas a co-occurrence edge is
  created when two words appear in the text
  within a word span of a specified length. The
  co-occurrence relation comes with two addi-
  tional parameters:

  - **Window size**: Maximum number of
    words constituting the span.

  - **Window application**: The window can
    be applied before or after filtering away
    tokens of unwanted POS, i.e. we can re-
    quire either the co-occurrence within the
    original text or in the filtered text.

- **Graph type**: A document can be represented
  as an unweighted or weighted graph. In
  the second case we use a weighted version
  of TextRank algorithm (Mihalcea and Tarau,
  2004) in which the strength of a vote depends
  both on the rank of the casting node and on
  the weight of the link producing the vote.

  - **Edge weights**: In general, the weight
    of an edge between any two nodes de-
    pends on the number of identified rela-
    tions, but we also experimented with as-
    signing different weights depending on
    the relation type.

- **Normalisation**: This parameter refers to nor-
  malising word ranks computed for the longer
  and the shorter text unit.

- **Word limit**: The maximum number of top-
  ranked words included in vector representa-
  tion of the longer text. May be equal to the
  number of words in the shorter of the two
  compared texts, or fixed at some arbitrary
  value.

3.2.3 Semantic extension parameters
The following factors regulate the impact of addi-

tional words retrieved from WordNet:

- **Synset limit**: The maximum number of
  synsets (word senses) retrieved from Word-
  Net per each word. Can be controlled by
  word ranks returned from the summariser.

- **Synonym limit**: The maximum number of
  synonyms (per synset) added to vector repre-
  sentation of the document. Can be controlled
  by word ranks.

- **WordNet relations**: The types of semantic
  and lexical relations used to acquire addi-
  tional synsets.

3.2.4 Similarity comparison parameters
- **Bag-of-words model**: The type of bag-of-
  word used for cosine comparisons.

- **Semantic match weights**: The rules for ad-
  justing weights of WordNet words that match
  observed words from the other vector.

3.3 Results
The above parameters in various combinations
were applied in an extensive series of experiments.
Contrary to our expectations, the results indicate
that the summariser has either no impact or has a
negative effect. Table 1 presents the set of param-
eters that seem to have impact, and the values that
resulted in best scores, as calculated by SemEval
Task 3 evaluation tool against the training dataset.

3.3.1 Discussion
In the course of experiments we consistently ob-

served higher performance when all words from
both compared documents were included, as op-
posed to selecting top-ranked words from the
longer document. Furthermore, less restrictive cri-
teria for merging tended to give better results.
We noticed clear improvement after extending word vectors with synonyms and related words. WordNet relations that contributed most are similar to, hypernym (ISA relation), pertainym (relational adjective) and derivationally related form. The results obtained before and after applying summarisation and lexico-semantic expansion (while keeping other parameters fixed at values reported in Table 1) are shown in Table 2.

| Parameter             | Value                  |
|-----------------------|------------------------|
| Word limit            | no limit               |
| POS                   | JJ, NN, V              |
| Merging criteria      | lemma, ignore POS      |
| Custom stopword list  | yes                    |
| Synset limit          | 15                     |
| Synonym limit         | no limit               |
| WordNet relations     | similar to, pertainym, hypernym |
| Bag-of-words model    | binary                 |

Table 1: Parameter values yielding the best scores.

| Word ranks          | Expansion | No   | Yes  |
|---------------------|-----------|------|------|
| Ignored             |           | 0.728| **0.755** |
| Used to select top-rank words | | 0.690 | 0.716 |
| Used to control synset limit | | N/A  | 0.752 |
| Used to weight vector counts | | 0.694 | N/A  |

Table 2: The effects of applying text summarisation and lexico-semantic expansion.

The results benefited from adding synonyms and semantically and lexically related words. Lemmatisation and merging same-lemma words regardless the POS, as well as ignoring very general verbs seem to be helpful.

| Category   | Test data | Train data |
|------------|-----------|------------|
| newswire  | 0.907     | 0.926      |
| cqa        | 0.778     | 0.779      |
| metaphoric | 0.099     | -0.16      |
| scientific | 0.856     | -          |
| travel     | 0.880     | 0.887      |
| review     | 0.752     | 0.884      |
| overall    | 0.785     | 0.755      |

Table 4: Break-down of the results.

4 Conclusions

We described our approach, parameters used in the system, and the results of experiments. Text summarisation didn’t prove to be helpful. One possible explanation of the neutral or negative effect of summarisation is the small size of the texts units: with the limited number of words available for comparison, any procedure reducing this already scarce set may be disadvantageous.

The results benefited from adding synonyms and semantically and lexically related words. Lemmatisation and merging same-lemma words regardless the POS, as well as ignoring very general verbs seem to be helpful.

The best performance has been observed in Newswire category. Finally, given that the similarity metric used by the system is essentially a
function of word overlap between the two texts, it is not surprising that the system struggled with metaphorically related texts.

References

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

Christiane Fellbaum. 1998. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

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)*. August 23-24, 2014, Dublin, Ireland.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of EMNLP 2004*:404–411, Barcelona, Spain.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, Vol. 38, No. 11:39–41.