Probabilistic models for sentence level similarity

Daniel Rižnar, Uroš Kosič

University of Ljubljana
Faculty of Computer and Information Science
Tržaška 25, 1000 Ljubljana
{driznar, ukosic}@gmail.com

***Abstract*—Sentence similarity measures became increasingly important in text-related research and application areas such as text mining, information extraction, automatic question-answering, text summarization, text classification and machine translation. In this work, we present an overview of existing methods and models used for this purpose. We implemented two models: probabilistic model with expanded text representation and query likelihood ranking function using Dirichlet smoothing with added synonyms. We provide results of experiments with these models. Our measures are based on 65 sentence pairs from Pilot Short Text Semantic Similarity Benchmark Data set. Results show, that probabilistic model with expanded text representation performs better than query likelihood ranking function using Dirichlet smoothing, when sentences share only topical similarities. Both methods perform better than simple word overlap method.**

***Index Terms*—expanded text representation, probabilistic models, sentence similarity, translation models**

I. Introduction

Measuring the similarity between documents and queries has been thoroughly studied in articles related to information retrieval. However, as areas of text mining, information extraction, summarization and machine translation spread, a need of computing the similarity between two very short segments of texts (sentences) has emerged. Unfortunately, standard text similarity measures perform poorly on such tasks because of data sparseness and the lack of context. Such measures rely heavily on terms occurring in both sentences. If they have few terms in common, the score will be low. Because these methods don't use topical relations between the sentences, they suffer from vocabulary mismatch problem. In sentence level similarity measures the topical relation is very important, but hard to acquire. For example “USA” and “United States of America” are semantically equivalent, yet share no terms in common.

Another problem when measuring the similarity between two sentences, is lack of context information. Standard methods rely on reasonable amount of text in a document, while sentences provide only a limited context. For example, “Apple computer” and “apple pie” share 50% of their terms, but are topically different. Standard methods would rate these two segments as similar, although topically, they are completely different. If we add additional document describing “apple pie”, the results are much better.

In Section 2, we provide an overview of related work, with emphasis on probabilistic models. We examine several approaches to sentence level similarity:

- *Word overlap measures*
  This is a baseline measure, measuring simple overlap of words.
- *TF-IDF measures*
  These measures are a broad class of functions used for estimating relevance and similarity topically between queries and documents.
- *Relative-frequency measures*
  These measures have been shown to perform well at finding co-derivative documents.
- *Probabilistic models*
  Translation transforms text in one language to text in another, with the aim of preserving as much of the semantics as possible. In [2] they propose using statistical translation models in much the same manner to estimate the probability that one sentence is a translation of another.
  In [1] they present probabilistic method with real time expanded representation of sentences. We use this method later in out experimental research.

In Section 3, we present our evaluation experiments and details about data used in measures. We chose data from Pilot Short Text Semantic Similarity Benchmark Data set. Data set contains 65 sentence pairs, already rated by 32 human participants. This data set is widely used as a benchmark for validating short text semantic similarity.

In Section 4, we wrap up and provide conclusions and directions for future work.

At the end we added acknowledgments and references.

II. Related Work

There is a large literature on probabilistic approaches to information retrieval. However, when we want to measure similarity between two sentences, these methods don't perform very well.

In [2] they focused themselves on intermediate levels of similarity. They explored mechanisms for measuring such intermediate kinds of similarity, focusing on the task of identifying where a particular piece of information originated. Their main topic of research was tracking information flow through various texts. Within their work, they provided some methods for measuring similarity on a sentence level. We will discuss them briefly in following passage.

All their techniques calculate similarity score S(Q,R) between sentences Q and R, intended to capture numerically, the extent to which they convey the same information. All methods return maximized S(Q,R), when sentence R has highest degree of similarity to the sentence Q.

A. Word overlap measures

They chose this measure as a baseline measure. Word overlap means the proportion of words in Q that also appear in the candidate sentence R:

$$S(Q,R)=\frac{|Q\cap R|}{|Q|}$$

$|Q\cap R|$ is the number of terms that appear in Q and R
$|Q|$ is the number of all terms that appear in Q

The logic behind word overlap is simple – if two sentences have many terms in common then they are probably similar to some degree.

They also experimented with adjusted version of word overlap, where they took inverse document frequency (IDF) into account. IDF is actually a weight used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

$$S(Q,R)=\frac{|Q\cap R|}{|Q|}\sum_{w\in Q\cap R}\log\left(\frac{N}{df_w}\right)$$

N is total number of documents in the corpus
df_w is number of documents where the term w appears.

Apparently, this adjustment requires additional context (in their terminology additional documents). We will discuss obtaining this additional data later in this section (we call that expended representation of a sentence).

B. TF-IDF measures

Term frequency-inverse document frequency (tf-idf) is actually a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or a corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. The basic intuitions are that the more frequently a word appears in a passage, the more indicative that word is of the topicality of that passage; and that less frequently a term appears in a collection, the greater its power to discriminate between interesting and uninteresting passages.

$$S(Q,R)=\sum_{w\in Q\cap R}\log(tf_{w,Q}+1)\log(tf_{w,R}+1)\log\left(\frac{N+1}{df_w+0.5}\right)$$

$tf_{w,Q}$ is number of times term *w* appears in sentence *Q*
$tf_{w,R}$ is number of times term *w* appears in sentence *R*
N is the total number of documents in the collection
df_w is the number of documents that *w* appears in

First two logs measure frequency of a term in both sentences, so bigger number means greater participation of a term in both sentences. Last log is an inverse document frequency, used to lower the score for common, unrepresentative words, such as: the, an, a, be, so...

C. Relative-frequency measures

Relative-frequency measures have been shown to perform well at finding co-derivative documents. In their work, they investigate how well such methods work at finding co-derivative pieces of text at the sentence level.

$$\frac{1}{1+\frac{max(|Q|,|R|)}{min(|Q|,|R|)}}\sum_{w\in Q\cap R}\frac{\log\left(\frac{N}{df_w}\right)}{1+|tf_{w,Q}|-|tf_{w,R}|}$$

$max(|Q|,|R|)$ is the length of longer sentence
$min(|Q|,|R|)$ is the length of shorter sentence
Other quantities are defined as above.

The numerator is a standard IDF factor explained above, while the denominator consists of two parts. First part (outside the sum) penalizes differences in the overall lengths of the sentences. Second part (inside the sum) penalizes inequalities in the relative frequency of a word between the two sentences.

D. Probabilistic models

Probabilistic models are based on idea of estimating the probability that one sentence is a translation of another. This translation probability then serves as the basis of the similarity score for pairs of sentences. Statistical machine translation systems aim to generate high-quality translations of sentences between natural languages. Such systems make use of parametrized statistical language models of both source and target language, and a parametrized statistical translation model that estimates the probability that a given target sentence is a translation of the source sentence. Given these models and a parametrization, the system searches a space of possible translations and returns the sentence with the highest probability. In their paper, they propose using statistical translation models in much the same manner to estimate the probability that one sentence is a translation of another. However, as our problem is different from normal translation problems (both sentences are in the same language), we can make some assumptions. We will now briefly summarize their path from more general model to a model adequate to our problem. We will also add some motivation and description of some specific terms.

They start with IBM's Translation Model 1. IBM Model 1 is a generative model. Generative modeling means breaking up the process of generating the data into smaller steps, modeling the smaller steps with probability distributions, and combining the steps into a coherent story ([5]). They provide following similarity function, based on IBM model 1:

$$S(Q,R)=\frac{1}{(|R|+1)^{|Q|}}\prod_{i=1}^{|Q|}\sum_{j=1}^{|R|+1}P_t(q_i|r_j)$$

$|R|$ is the length of sentence R

$|Q|$ is the length of sentence Q

$P_t(q_i|r_j)$ is a probability that j-th word in R is a translation of i-th word in q

Then they made some additional assumptions. The original model assumes that each sentence has a special *null* term at position 1; this is the reason that the summation iterates through |R| +1 terms. The null term is used to represent the fact that the current term in Q doesn't align to any terms in R.

With that in mind, they make the distributional assumption that $P_t(q_i|r_1)=P(q_i|C)$, where C is the background model inferred from the collection as a whole. This proceeds from the intuition that - in the absence of any other evidence – an unaligned word is likely to be present in a sentence with a probability equal to its overall probability in the more generalized background language model. The probability of aligning to the null term dictates the influence of the background language model on the resulting translation. Because IBM Model 1 assumes that all reorderings are equally likely, the probability that a term in Q will align to the null term is $\frac{1}{|R|+1}$. Then they generalize the original model by assuming there exists μ null terms in each sentence, where μ is a non-negative integer. This results in each sentence having length |R| + μ, where |R| is the number of non-null terms in R. This model can be described as:

$$S(Q,R)=\frac{1}{(|R|+\mu)^{|Q|}}\prod_{i=1}^{|Q|}\left[\sum_{j=1}^{\mu}P(q_i|C)+\sum_{j=\mu+1}^{|R|+\mu}P_t(q_i|r_j)\right]$$

μ is the number of null terms in each sentence

$P(q_i|C)$ is a probability that i-th term in Q appears in some background model C

They simplify the model further, with assuming that each word translates to itself; that is $P_t(q_i|r_j)=1$ if $q_i=r_j$. This results in the following form:

$$S(Q,R)=\prod_{i=1}^{|Q|}\frac{tf_{q_i,R}+\mu P(q_i|C)}{|R|+\mu}$$

$tf_{q_i,R}$ is the frequency of i-th word in sentence Q in sentence R

Above function is known as language modeling *query likelihood ranking function* using Dirichlet smoothing parameter μ. With μ=1, we get Berger and Lafferty's Translation Model 0. All models here assume that every term only translates to itself. We extended this model with synonyms and so incorporated a more refined estimate of the true translation probabilities. As parameter μ approaches 0, the model becomes word overlap measure that will likely be good at finding exact matches. At the other extreme, as μ gets large more background terms are allowed, which is likely (and known to be) good at finding topically relevant matches.

They defined similarity spectrum, where at one end there is exact identity and at the other general topic relation. They divided this spectrum into 5 parts: exact match, minor edit, same facts, specific topic match, general topic match. They found out that at the general and specific topic level, query likelihood function with μ=2500 gives the best results. This was expected, because past research has shown query likelihood to be effective at identifying topicality. At other levels the relative performance difference between techniques was small, but Translation Model 0 (μ=1) was consistently the most effective.

E. Negative KL-Divergence and expanded representation

In [1] they investigated different similarity measures for short segments of text. They also took different text representation into account:

- surface representation is the most basic representation of a short text segment – the text itself.
- stemmed representation is normalization of a text (ie.

"marine vegetation" becomes "marin veget"). Although stemming can significantly improve matching coverage, it also introduces noise, which can lead to poor matches.

- expanded representation is good for handling contextual problems. If stemming fails to discern the difference between the meaning of "bank" in "Bank of America" and "river bank", expanded representation can gather additional contextual information and perform better in these cases. One approach is to enrich the representation using an external source of information related to query terms. Possible sources of such information include web search results returned by issuing the short text segment as a query, relevant Wikipedia articles... Each of these sources provides a set of contextual text that can be used to expand the original sparse text representation. In their experiments, they used web search results to expand short text representation.

They used this expanded representation with probabilistic measure framework. They define the problem as: given two short segments of text, Q and C, treating Q as the query and C as the candidate we wish to measure similarity of. For ranking purposes they used the negative KL-divergence between query and candidate model.

$$-KL(\Theta_Q,\Theta_C)=H(\Theta_Q)-CE(\Theta_Q,\Theta_C)\equiv\ldots$$
$$\ldots\equiv\sum_{w\in V}P(w|\Theta_Q)\log(P(w|\Theta_C))$$

$H(\Theta_Q)$ is the entropy for the query model

$CE(\Theta_Q,\Theta_C)$ is the cross entropy for query and candidate models

V is the vocabulary (all unique words in both sentences Q and C)

$P(w|\Theta_Q)$ is the estimation of query model

$P(w|\Theta_C)$ is the estimation of candidate model

These estimates are defined as:

$$P(w|\Theta_Q)=\frac{tf_{w,QE}+\mu_Q P(w|C)}{|QE|+\mu_Q}$$

$tf_{w,QE}$ is the frequency of term w in QE

μ_Q is Dirichlet smoothing parameter

$P(w|C)$ is the probability of word w in candidate sentence C

QE is the query sentence Q expanded representation

$$P(w|\Theta_C)=\frac{tf_{w,CE}+\mu_C P(w|C)}{|CE|+\mu_C}$$

$tf_{w,CE}$ is the frequency of term w in CE

CE is the candidate sentence C expanded representation

These estimations are basically the same as query likelihood ranking function presented on previous page.

Their results show that probabilistic methods are good at finding topicality related matches. The probabilistic framework presented in their paper provides a general method for measuring the similarity between two short segments of text.

III. Results

In this section we present methods and models used in our experiments, data used in our measures and evaluation of these methods.

A. Methods

We implemented two of previously described probabilistic methods. First one is query likelihood ranking function using Dirichlet smoothing parameter, with taking synonyms into account.

$$S(Q,R)=\prod_{i=1}^{|Q|}\frac{tf_{q_i^S,R}+\mu P(q_i^S|C)}{|R|+\mu}$$

q_i^S is now a set of words – original word plus all its synonyms

Other variables are defined as above.

Synonyms are found with Java API for WordNet Searching (JAWS). For example, a sentence "An automobile is a car." is expanded to "An {car, automobile, auto, motorcar, machine} is a {car, automobile, auto, motorcar, machine}.

The other implemented model is Negative KL-Divergence with expanded representation of sentences. We expand our sentences similar as in [1]. We use a sentence as a query into Bing search engine to generate first 50 results. Then we take their descriptions and concatenate them, to get final expanded representation of a sentence. Because long sentences are usually very bad queries for web search engines, we first remove stop words from a sentence (a, about, that, the, then...) and form the search query only from nouns. Here is an example. We start with a sentence: "A cock is an adult male chicken. ". After removing stop words, we get a list of tokens [cock adult male chicken]. Then we extract the nouns with a Java machine learning toolkit for natural language processing (OpenNLP). We get: [cock adult chicken] and that is our search query. Here are first two results on Bing search engine:

- a cock is an adult chicken of the male sex. (In Britain, they call them rooster) a hen is an adult chicken of the female sex. a capon is an adult chicken of the neutered male

- cock 1 (k k) n. 1. a. An adult male chicken; a rooster. b. An adult male of various other birds. 2. A weathervane shaped like a rooster; a weathercock.

After concatenating first 50 descriptions we get some additional contextual information about the sentence.

B. Data

We used 65 sentences from [6]. Dataset was created as a benchmark for validating short text semantic similarity measures. Ratings used in the dataset follow the practice used in word similarity studies. Sentences were rated by humans and the "typical" human rating is the mean of those given by a set of participants. The measure of agreement is the Pearson product-moment coefficient quoted with statistical significance. The ratings are from a rating scale running from 0.00 to 4.00 (we scaled them onto interval from 0.00 to 1.00).

C. Results

All our implemented methods return values on interval from 0.00 to 1.00. KL-Divergence or Kullback-Leibler Divergence also known as information gain is a non-symmetric measure of the difference between two probability distributions P and Q. KL measures the expected number of bits required to code samples from P when using code based on Q, rather than using a code based on P. Typically P represents the "true" distribution of data, observations, or a precisely calculated theoretical distribution. The measure Q typically represents a theory, model, description, or approximation of P ([4]). Because it returns score in bits, we had to scale it to an interval from 0 to 1:

$$newVal = ((oldVal - fromMin)/(fromMax - fromMin)) \cdot ... \\ ... \cdot (toMax - toMin) + toMin;$$

oldVal is an old value we want to scale

fromMin is minimal value (that is the lowest score in the series)

fromMax is maximal value (that is the highest score in the series)

toMax is 1

toMin is 0

We also experimented with different values for smoothing parameter μ. As stated before, μ is basically the weight with which we glide between simple word overlap (low μ), capturing only syntactical similarities on one side and capturing some topical or contextual similarities on the other (high μ). The results are strongly influenced by quality of documents (or expanded representation) used for getting informations about topical similarities. We got the best results with μ=0.2, μ_Q=0.4 , μ_C=0.6.

The graphs bellow, present correlation between human scores provided with the dataset and our measurements with mentioned methods.

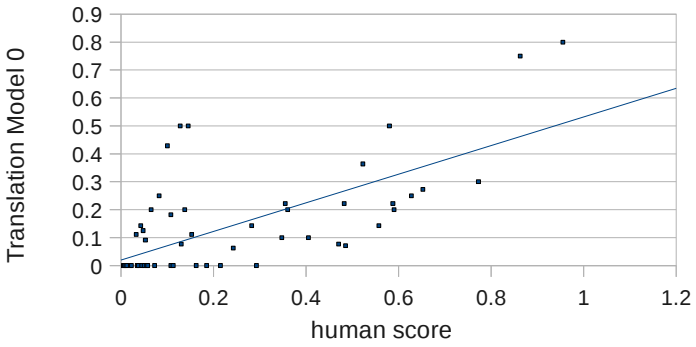

Fig. 1. For every sentence in dataset we plot human score (as given in the dataset) and score calculated with Query Likelihood method.

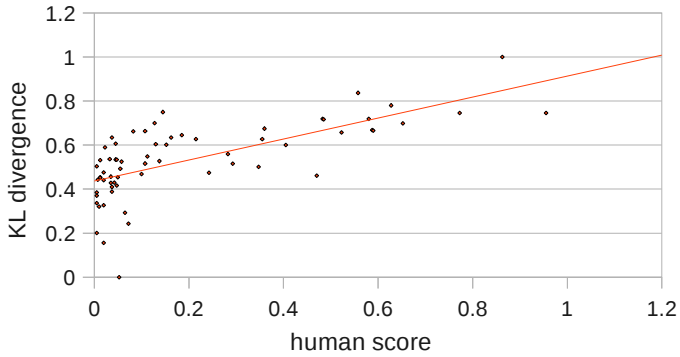

Fig. 2. For every sentence in dataset we plot human score (as given in the dataset) and score calculated with KL divergence method.

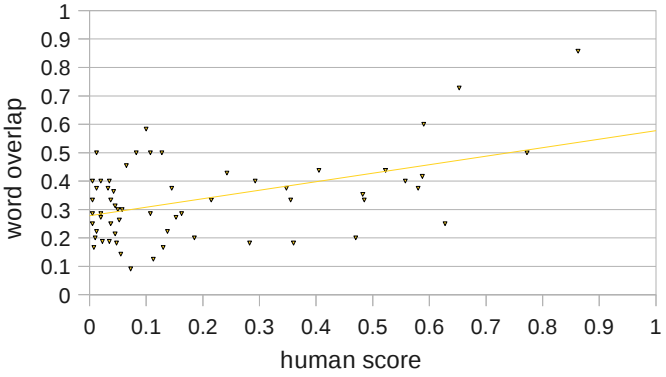

Fig. 3. For every sentence in dataset we plot human score (as given in the dataset) and score calculated with word overlap. Goal of our models is to get better than this.

We calculated the correlation coefficient between human scores and scores that our methods produced. We used following formula ([3]):

$$corr_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum x_y}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

Measure	**Correlation with human scores**
Word overlap	0.49
Translation model 0	0.64
KL-Divergence	0.68

Table 1. Correlation coefficients of our measures and human scores.

We can see, that both Query Likelihood Ranking function (QL) and KL-Divergence method perform better than Word overlap model.
Here are few interesting pairs of sentences used in experiments and their scores.

SENTENCE	HUMAN SCORE	KL-DIV. SCORE	QL SCORE
Midday is 12 o'clock in the middle of the day. Noon is 12 o'clock in the middle of the day.	0.95	0.73	0.77

Table 2. Most humans would rate these two sentences as very similar. Both our models rated the pair with high score.

SENTENCE	HUMAN SCORE	KL-DIV. SCORE	QL SCORE
A cemetery is a place where dead people's bodies or their ashes are buried. A graveyard is an area of land, sometimes near a church, where dead people are buried.	0.77	0.74	0.29

Table 3. Topically, these two sentences share some similarities. While KL-divergence with expanded representation manages to capture this, QL fails to recognize it.

IV. Conclusions and Future Work

In this paper we studied the problem of measuring the similarity between two sentences. We looked at various existing methods and models used for similar tasks – word overlap, tf-idf measures, relative-frequency measures, query likelihood ranking function with Dirichlet smoothing, negative KL-divergence. We showed how web search results can be used to form expanded representations of sentences. We then implemented two models: Query Likelihood ranking function with synonyms and negative KL-divergence with expanded representation. We then evaluated and compared these measures on prepared data set with 65 pairs of sentences and human ratings. It was shown that our methods perform better then simple word overlap and that expanding the representation of a sentence helps in finding topical similarities.

Although we chose to use web search results as the basis of our expanded representation in this work, it would be interesting if we could compare the results with some other types of representations: Wikipedia articles, topically related corpus of documents... There is also a possibility for research why and in what conditions Query Likelihood ranking function performs better than KL-divergence measure. It would also be interesting to see, how would these methods work in a practical application, in order to see what impact they have in a more practical setting.

Acknowledgment

We would like to thank our mentor, M.Sc Ercan Canhas for his help and time.

References

[1] D. Metzler, S. Dumais, and C. Meek, “Similarity Measures for Short Segments of Text”
[2] D. Metzler, Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel, “Similarity Measures for Tracking Information Flow”
[3] http://en.wikipedia.org/wiki/Correlation_and_dependence
[4] http://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence
[5] P. Koehn, *Statistical Machine Translation.* Cambridge: Cambridge University Press, 2010, ch. 4
[6] J. O'Shea, Z. Bandar, K. Crockett, and D. McLean, “Pilot Short Text Semantic Similarity Benchamrk Data Set: Full Listing and Description”, 2009