An Architecture for Scientific Document Retrieval Using Textual and Math Entailment Modules

Partha Pakray and Petr Sojka

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{pakray,sojka}@fi.muni.cz

Abstract. We present an architecture for scientific document retrieval. An existing system for textual and math-ware retrieval Math Indexer and Searcher MiAs is designed for extensions by modules for textual and math-aware entailment. The goal is to increase quality of retrieval (precision and recall) by handling natural language variations of expressing semantically the same in texts and/or formulae. Entailment modules are designed to use several, ordered layers of processing on lexical, syntactic and semantic levels using natural language processing tools adapted for handling tree structures like mathematical formulae. If these tools are not able to decide on the entailment, generic knowledge databases are used deploying distributional semantics methods and tools. It is shown that sole use of distributional semantics for semantic textual entailment decisions on sentence level is surprisingly good. Finally, further research plans to deploy results in the digital mathematical libraries are outlined.

Keywords: math-aware information retrieval, semantic textual entailment, math entailment, distributional semantics, Gensim

1 Introduction

Semantic-based document filtering and search module is a key component of any Information Retrieval (IR) system. Search is a gateway to the ever-growing database of documents in digital libraries (DL) or on the web. Even though keyword-based IR systems became part of everyday life today, they are not fully suitable for research search to DLs, for example. The more precise results the information seeker might get are those expressed, queried, indexed, and retrieved based on word, sentence, paragraph, or document meaning, e.g., semantic features of the document content.

The variation in expressivity of natural languages, including the mathematical vernacular, to describe semantically similar ideas and elements is enormous. Keyword-based information systems try to cope with it on lexical level by morphology (indexing lemmas) or by synonymical expansion like Wordnet. There is ‘semantic web’ and ontology-based approaches based on discrete, dichotomic representations of words and relations between them. But they are often not
enough to handle and uniformly represent document, paragraph, sentence or formulae meaning in IR systems, e.g. for semantically fine-grained document filtering and similarity computations.

On the other hand, distributional semantic approaches have deserved well-grounded attention recently. They allow to represent word or phrase meaning in continuous high-dimensional spaces, just based on unsupervised, and often deep, learning methods \[15\]. Such representations can be used for purposes like qualified guesses of semantic similarity of words, phrases, or even sentences or formulae.

In this paper, we design an extension module for our math-aware information system MIaS \[21\]. We argue that it will further increase current performance \[12,20\] by better, semantic clustering of variably expressed content.

The motivation for new architecture design is discussed in Section 2. We describe how distributional semantics may help to compute semantically similar text chunks or formulae. In Section 3 the new entailment modules of the architecture are described. We conclude by Section 4 by describing further directions of research.

2 Motivation for a New Architecture

When checking precision of MIaS on results from \[12,20\], we have realized that some documents are not found just because of minor rephrasing of formulae or text in query with respect to the document. We need a robust way of computing similarity for textual phrases and formulae terms. In STEM papers, the text is full of formulae, where we cannot simply discard them as they convey very important semantics in dense form: semantic textual similarity is needed.

2.1 Semantic Textual Similarity

The main goal of Semantic Textual Similarity (STS) task \[1\] is measuring the degree of semantic equivalence between a pair of texts, e.g. sentences. This task can applied in many areas as Information Extraction, Question Answering, Summarization and in Information Retrieval area for indexing the semantically same phrases or sentences. Three STS evaluation tasks were organised in 2012 \[3\], 2013 \[2\], and 2014 \[1\] at SemEval workshops. In that evaluation tasks, the systems performance was evaluated using the Pearson product-moment correlation coefficient between the participant system scores and the human scores.

Textual similarity problem may be tackled by various techniques at lexical, syntactic and semantic levels \[1\] as usual during NLP processing. Among lexical techniques there are word overlap metrics or \(n\)-gram matching. Another way is to compare dependency relations of two texts. In computations one can use synonyms, hypernyms, etc. The higher processing level, the better performance is usually achieved.
There always remain some examples which cannot be decided by lexical, syntactic nor semantical analysis, as full knowledge and meaning representation is needed for it. There is semantic gap between lexical surface of the text and its meaning because same concepts are represented in different vocabulary, languages, formalisms and notations. Updating knowledge databases with all dialectical possibilities in supervised way is doomed to failure.

In distributional semantics approaches [5], similarities between linguistic items could be computed from their collocativity and distributional properties in large samples of language data in unsupervised way, as clearly seen from visualization experiments [7]. Especially convincing are recent experiments computed by Gensim framework [18] where words and phrases are computed by Word2vec [14] language model. We have tried to use it for STS task.

2.2 Sentence Level Similarity Baseline Experiment

Our STS system will generate various kinds of features from each processing level as shown in Figure 1. Finally, it will use machine learning to decide on the similarity between two text chunks as shown in later on Figure 3 on page 113.

In a preliminary experiment we have used already pre-trained word and phrase vectors available as part of Google News dataset [14] (about 100 billion words). The LSA word-vector mappings model contains 300-dimensional vectors for 3 million words and phrases.

Gensim [18] is a Python framework for vector space modelling. We have used Gensim for this experiment, and computed the cosine distance between vectors representing text chunks – sentences from SemEval tasks.

equivalence. The SemEval organizers provided English sentence pairs of news headlines (corpus named HDL), pairs of glosses (OnWN), image descriptions (Images), DEFT-related discussion forums (Deft-forum) and news (Deft-news), and tweet comments and newswire headline mappings (Tweets).

Table 1: SemEval-2014 Task 10: Multilingual Semantic Textual Similarity Test Result

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Winner score and team/run name</th>
<th>Our score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deft-forum</td>
<td>0.5305 NTNU-run3</td>
<td>0.42812</td>
</tr>
<tr>
<td>Deft-news</td>
<td>0.7850 Meerakat_mafia-Hulk</td>
<td>0.67999</td>
</tr>
<tr>
<td>Headlines</td>
<td>0.7837 NTNU-run3</td>
<td>0.60985</td>
</tr>
<tr>
<td>Images</td>
<td>0.8343 NTNU-run3</td>
<td>0.71402</td>
</tr>
<tr>
<td>OnWN</td>
<td>0.8745 MeerkatMafia-paringWords</td>
<td>0.79135</td>
</tr>
<tr>
<td>Tweet-news</td>
<td>0.7610 DLS@CU-run1</td>
<td>0.76571</td>
</tr>
</tbody>
</table>

Table 2: SemEval-2013 Task 6: Semantic Textual Similarity Test Result

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Winner score and team/run name</th>
<th>Our score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headlines</td>
<td>0.7838 UMBC_EBIQUITY-saiyan</td>
<td>0.62501</td>
</tr>
<tr>
<td>OnWN</td>
<td>0.8431 deft-baseline</td>
<td>0.71165</td>
</tr>
<tr>
<td>FNWN</td>
<td>0.5818 UMBC_EBIQUITY-ParingWords</td>
<td>0.38353</td>
</tr>
<tr>
<td>SMT</td>
<td>0.6181 UMBC_EBIQUITY-ParingWords</td>
<td>0.32951</td>
</tr>
</tbody>
</table>

Table 3: SemEval-2012 Task6: Semantic Textual Similarity Test Result

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Winner score and team/run name</th>
<th>Our score</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRpar</td>
<td>0.6830 baer/task6-UKP-run2_plus_postprocessing_smt_twsi</td>
<td>0.30103</td>
</tr>
<tr>
<td>MSRvid</td>
<td>0.8803 jan_snajder/task6-takealab-simple</td>
<td>0.68318</td>
</tr>
<tr>
<td>SMT-europal</td>
<td>0.5581 sranjans/task6-sranjans-1</td>
<td>0.54057</td>
</tr>
<tr>
<td>On-WN</td>
<td>0.7273 weiweitask6-weiwei-run1</td>
<td>0.68779</td>
</tr>
<tr>
<td>SMT-news</td>
<td>0.6085 desouzatak6-FBK-run3</td>
<td>0.51915</td>
</tr>
</tbody>
</table>

Tables 1, 2, and 3 show results of our minimalistic system based on distributional semantics language model compared to highest sentence similarity scores of systems participating in SemEval-2014, 2013 and 2012. It is worth noting that for Tweet-news subtask at SemEval-2014 our ‘baseline’ system using only plain Word2vec with pretrained Google news data by LSA gave better result than the best system at SemEval-2014!

Just recently, another way of computing global distributional semantics has been reported by Stanford’s GloVe [16]. We will compare its performance with Word2vec. As our results on SemEval data indicate that training corpora is very important, we have realized that Wikipedia knowledge to tackle the
STS Similarity problem is crucial, including the named entities and formulae available there.

2.3 Learning from Wikipedia Corpus

Wikipedia is an online encyclopedia that contains millions of articles on a wide variety of topics with quality comparable to that of traditional encyclopedias. In [22, 17, 23], Wikipedia has been used as a successful measure of semantic relatedness between words or text passages.

We will build word and phrase vectors from Wikipedia articles[1]. This Wikipedia dump contains more than 3 billion words. We will use Word2vec for learning high-quality word vectors from Wikipedia data sets with billions of words. An example for vector representation could be as follows: vector("King") − vector("Man") + vector("Woman") results in a vector that is closest to the vector representation of the word Queen. [15]

We will test on SemEval STS test data by using this generated vector from Wikipedia articles. Finally, we will compare our results with our baseline system. We will also participate in STS evaluation track at SemEval 2015 Task 2[2]. Having good similarity measures on scientific text chunks, we may use it for our math-aware information retrieval system.

3 New MIaS Architecture with Entailment Modules

Our top-level system architecture is shown in Figure 2. The architecture used sofar is enriched by three modules: Text-Text Entailment (TE), Math-Math Entailment (ME) and Text-Math Entailment (TME) modules.

Textual entailment is defined in [9] as: text \( T \) is said to entail hypothesis \( H \) if the truth of \( H \) can be inferred from \( T \). The task of Textual entailment is to decide whether the meaning of \( H \) can be inferred from the meaning of the \( T \).

For example, the text \( T = "John’s assassin is in jail" \) entails the hypothesis \( H = "John is dead"; \) indeed, if there exists one’s assassin, then this person is dead. On the other hand, \( T = "Mary lives in Europe" \) does not entail \( H = "Mary lives in US". \) Much effort is devoted by the Natural Language Processing (NLP) community to develop advanced methodologies in TE which is considered as a core NLP task. Various international conferences and several evaluation track competitions on TE have been held, notably at PASCAL-Pattern Analysis, Statistical Modelling and Computational Learning[3], Text Analysis Conferences (TAC)[4] organized by the United States National Institute of Standards and Technology (NIST), Evaluation Exercises on Semantic Evaluation (SemEval)[5], National Institute of Informatics Test Collection for Information Retrieval

1 http://dumps.wikimedia.org/enwiki/
2 http://alt.qcri.org/semeval2015/task2/
3 http://pascallin.ecs.soton.ac.uk/Challenges/
4 http://www.nist.gov/tac/tracks/index.html
5 http://semeval2.fbk.eu/semeval2.php
System (NTCIR\textsuperscript{6}) since 2005. At each new TE competition, the participating teams introduced several new features in their TE systems ranging from lexical to syntactic to semantic methodologies from two-way (i.e. binary-class) to multi-way (i.e. multi-class) textual entailment classifications in monolingual to cross-lingual scenario in order to solve the TE problem.

In this work we will investigate into the use of entailment modules for IR. We will show that Textual and Math entailment plays a significant role for monolingual IR performance.

The general architecture of Textual Entailment system is shown in Figure \textsuperscript{3} on the next page. Text and Hypothesis comparison is represented by comparative analysis; and the entailment decision is made by a classifier that makes use of a feature vector.

The Textual Entailment system is unidirectional but Semantic Textual Similarity is mainly bidirectional. Table \textsuperscript{4} on page 115 shows our system result of Semantic Textual Similarity and compare to the Entailment.

In the MIaS system \textsuperscript{21} search can be done by three ways e.g. only text search, only mathematics formula search and text with mathematics formula
search. During the searching phase, a query can match several terms in
the index. However, one match can be more important to the query than
another, and the system must consider this information when scoring matched
documents. An example of TE module is shown in Figure 4 on the next page.

ME module will compare between Math query and document that con-
tained math formula. For example, \( x^2 + y^2 = z^2 \) entails \( a^2 + b^2 = c^2 \). We will
implement Math Entailment in Formulae weighting module [21]. We will try to
use Math Entailment module in this phase to find appropriate terms. An exam-
ple of the ME module is shown in Figure 5 on the following page.

TME Module will compare text and math within documents. TME module
not only increases fairness of similarity ranking, but also helps to match a query
against the indexed form by adding new terms for indexing, e.g. formulae for
named entity used to name it. TME module is shown in Figures 4 and 5 on the
next page.

Entailment module will search not only for whole sentences (whole formu-
lae), but also for single words and phrases (subformulae down to single var-
iables, symbols, constants, etc.). For calculating the relevance of the matched ex-
pressions to the user’s query, entailment module will use a matching technique
of indexed mathematical terms, which accordingly affects scores of matched
documents and thus the order of results.

In our TE system based on lexical similarity we will determine the similarity
between the two texts by our STS module. Additionally, we will compare the
dependency structure between the two texts.

The TE problem can be tackled by various ways like lexical, syntactic and
semantic. Sometimes lexical semantic similarity is not sufficient to solve the TE
problem. In Table 1 for pair Id 5 our lexical semantic similarity system have
given high score of 0.95 but the meaning of text1 and text2 is very different.
In this case dependency structure weighting verb as main decision factor may
solve the problem.

Tree structure of input sentences are widely used by many research groups,
since it provides more information with quite good robustness and runtime
than shallow parsing techniques. Basically, a dependency parsing tree contains
nodes (i.e., tokens/words) and dependency relations between nodes. Some approaches simply treat it as a graph and calculate the similarity between the text and the hypothesis graphs solely based on their nodes, while some others put more emphasis on the dependency relations themselves. The recent approaches of syntactic or tree edit models are [10][13][19]. The approach in [11] based on the tree edit distance algorithm, which contains three basic operators, insertion, deletion and substitution. Insertion is defined as the insertion of a node from the dependency tree of $H$ into the dependency tree of $T$; deletion is the removal of a node from the dependency tree of $T$, together with all its attached children; and substitution is the change of the label of a node.
Table 4: Example of pairs from Task 1 at SemEval 2014

<table>
<thead>
<tr>
<th>Id</th>
<th>Text 1</th>
<th>Text 2</th>
<th>our STS</th>
<th>Entailment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One young boy is climbing a wall made of rock</td>
<td>A young child is climbing a rock climbing wall which is indoors</td>
<td>0.7871</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>A man is phoning</td>
<td>A man is talking on the phone</td>
<td>0.8238</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>John was born on January 15, 1986 in Kolkata.</td>
<td>John was born in 1986 in the city of Kolkata.</td>
<td>0.7996</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>A woman is performing a trick on a ramp with a bicycle</td>
<td>A woman is jumping with a bicycle</td>
<td>0.7839</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>A brown dog is attacking another animal in front of the man in pants</td>
<td>A brown dog is helping another animal in front of the man in pants</td>
<td>0.95</td>
<td>No</td>
</tr>
</tbody>
</table>

in the source tree (the dependency tree of $T$) into a label of a node of the target tree (the dependency tree of $H$). Substitution is allowed only if the two nodes share the same part-of-speech (POS). The approach in [4] presents a new data structure, termed compact forest, which allows efficient generation and representation of entailed consequents, each represented as a parse tree. Rule-based inference is complemented with a new approximate matching measure inspired by tree kernels, which is computed efficiently over compact forests. The approach [24] built a model to solve the entailment problem by using dependency syntax analysis (by Stanford Parser), lexical knowledge base (e.g. WordNet), web information (e.g. Wikipedia) and probabilistic methods.

We will generate dependency tree for two texts. Then mapping can be done in two ways e.g. directly (when entities from hypothesis dependency tree exist in the text tree) or indirectly (when entities from text tree or hypothesis tree cannot be mapped directly and need transformations using external resources). Based on this step we will decide on our entailment resulting implementation.

4 Conclusion and Further Work

We have described an architecture for math-aware information retrieval that employs textual and math entailment. We have described our further research directions: distributional approaches that we will test for entailment modules. We want also train distributional semantics representation for mathematical formulae, and test to which extent their vectors may be used to approximate their meaning. Finally, we plan to use SEPIA evaluation tool and NTCIR’s Math task [12] data to evaluate the improvements, and eventually use it in the digital mathematics libraries as EuDML [6] or planned GDML [8].

Acknowledgement This work was supported by an ERCIM Alain Bensoussan Fellowship 2014–15 and Masaryk University. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the view of the ERCIM or MU.
References


