Siena’s Tasks Track System

Emily Barranca
Swathmore College
500 College Avenue
Swathmore, PA 19081
ebarran1@swarthmore.edu

Sharon Gower Small
Siena College Institute for Artificial Intelligence
515 Loudon Road
Loudonville, NY 12211
ssmall@siena.edu

Abstract

This paper discusses our development of Siena’s Tasks Track System (STTS) and its participation in the Tasks Track of the 2017 Text Retrieval Conference (TREC). The purpose of this track is to try to understand the underlying task a user might be trying to complete when searching a specific query. The Task Understanding portion of this task aimed to find a list of up to 100 keywords and phrases that might be related to a user’s task given the example query. In the first year of our work in this track we submitted two fairly simple runs. In the first run, we used the jsoup library to run these queries through Google and processed the first five documents resulting from the search. We automatically extracted keywords and phrases that appeared commonly across the documents. In the second run we utilized Wordnet to generate new words and phrases to augment the query. These two methods, the Wordnet approach and the Google approach were submitted as two separate runs.

1. Introduction

The Tasks Track (Verma et. al., 2016) is a track in the Text Retrieval Conference (Voorhees, 2007), hosted by the National Institute of Standards and Technology. The Tasks Track is one track in the TREC conference, which aims to understand the larger infromation task a user might be attempting to accomplish when entering a single query into a search engine. In the completion of the Task Understanding piece of this track, participants were to produce a list of up to 100 ranked keywords/phrases for each of 50 queries.

2. TREC 2017 Literature Review

While designing the experimental procedure for this year’s task track, our team reviewed literature from the 2015 and 2016 Task Track submissions. The Microsoft Research team produced a list of ranked keywords using anchor text to grab links where the destination is an external site (Bennett & White, 2015). Similarly in 2016 at Bauhaus University in Weimar, Germany, their team used anchor text to generate keyword queries under “the assumption that the texts in HTML anchors (with href attribute) are short descriptions of the documents they link to” (Hagen et. al, 2016). The Bauhaus University team also submitted a run consisting of related search suggestions from the common search engines Bing and Google, as they had done for the previous year’s submission.

In 2016, the University of Stavanger team (Garigliotti, Balog, 2016) used keywords extracted from relevant documents and search engine suggestions, taken from Google and

Bing. In their procedure, the top ten documents were collected from Google and Bing, and then they applied the RAKE keyword extraction system, which first isolates all possible words and phrases and then formulaically and using machine learning, ranks the likelihood a word is a keyword. Their team then formulaically considered the success of adding the keywords found to the beginning of a given query, the end, or whether the keyword stood best on its own

3. Main System Components

To complete the Task Understanding problem of the 2017 Tasks Track, our team submitted two runs, the first of which was focused on extracting keywords and phrases from the top N documents (N=5) returned by Google for each given query. The second run submitted by our team focused on the use of Wordnet (Miller, 1995) software to find keywords or phrases that can be used to augment the given queries.

3.1 Run 1: Topic Detection

3.1.1 Document Processing

In order to retrieve the top documents returned by Google, our team used the open source jsoup[1] Java library, a parser for working with real world HTML. Using a test set of previous years' results we ran experiments to determine the ideal number of documents to process and the ideal window of text to process within those documents.

After experimentation of document ranges between 2 and 15 we found that the first 5 documents of the Google search gave the best results when processed. We also found, for our test set, that the first 1000 characters from each document is the ideal set of text to process. From this text we removed common stopwords. We also generated a custom list of stopwords that were common in webpages, not filtered by the jsoup html to text method, and not useful for the purposes of this task, i.e. *site map, site content, english version,* etc. This list certainly could have been longer and more encompassing, but in the time frame we had we were able to generate 192 stopwords and phrases by manual analysis of results from the test set.

3.1.2 Common Phrases

Our next step was to find the most common overlapping noun phrases between the top 5 documents in their first 1000 characters. Based on previous research (Small & Strzalkowski, 2004) in topic generation we hypothesized that these overlapping noun phrases would generate a good list of topics related to the queries. We used the Stanford part of speech tagger (Toutanova & Manning, 2000) to tag the text. We then used this tagged text to generate a list of the noun phrases that existed in each block of text and then looked for the most commonly occurring phrases in this set, with longer phrases taking precedence over shorter phrases. For this task we returned the top 20 phrases found, ordered by frequency as the primary metric and number of words in a phrase as a

[1] https://jsoup.org

secondary metric. That is words/phrases are ordered based on frequency first and within these they are ordered based on number of words in each phrase.

For each of the topics we submitted our top 20 keywords and phrases to be evaluated. One of our better performing topics, topic 8 is shown below. We scored a .97 nDCG@20 for this topic.

TREC Topic:

```
<task id = "8">
<query>time travel</query>
<freebase entity = "1">
<id>/m/07s2s</id>
<text>time travel</text>
</freebase>
</task>
```

STTS Output in Ranked Order:

1 other uses time travel
2 time machine travel
3 certain points time travel
4 quantum mechanics time travel
5 different points time travel
6 time travel
7 hypothetical device time travel
8 time dilation travel
9 distant points time travel
10 arbitrary point time travel
11 limited support time travel
12 theoretical physics time travel
13 Einstein-Rosen bridges time travel
14 narrow sense time travel
15 well-understood phenomenon time travel
16 special relativity time travel
17 general relativity time travel
18 large amount time travel
19 current technology time travel
20 possible Russian Cosmonaut time travel

3.2 Run 2: Wordnet on Query Phrases

For our second run, we hypothesized that useful phrases could be generated by finding synonyms or other related words to replace particular words in the given queries. Again, for our second run, we tested our procedure with sample queries from previous years to optimize our output. This process however did not always return a variation of 20 new phrases. When our Run 2 did not have 20 new phrases generated we supplemented its list with the phrases from Run 1.

3.2.1 Wordnet MIT Java Wordnet Interface

To access the Wordnet software in our design, we used the MIT Java Wordnet Interface Library (Finlayson & Alan, 2014). Our team found synonyms for each word in the query and found synsets for each verb in each query. Synsets are defined by Wordnet to be "*words that denote the same concept and are interchangeable in many contexts*".

3.2.2 Related Verbs

After analyzing the sample queries we worked with, we found that the nouns in the query were typically most vital to the user's task and desired information. The verbs, however, when replaced by synsets and related words, did not result in loss of meaning in the user's task. To distinguish which words from the queries were verbs and which words were nouns, we ran each query through the Stanford Part of Speech tagger.

3.2.3 Generating Key Words

Our new keywords/phrases were generated by switching out the related words returned to us by Wordnet with each verb in each query. Additionally, we replaced nouns and verbs with their synonyms, only changing one word at a time from the original query, as to not lose the user's original task. Our highest ranked output words were the phrases generated by replacing verbs with related words, then with synonyms. Lastly, came our results generated by replacing nouns from the original query with synonyms.

Shown below are the generated key words for topic 23. For this run we did not have to augment with any phrases from Run 1. For Run 2, these words received a .83 nDCG@20 score.

TREC Topic:

```
<task id = "23">
<query>put a newborn baby to sleep</query>
<freebase entity = "1">
<id>/m/0jnvp</id>
<text>newborn baby</text>
</freebase>
<freebase entity = "2">
<id>/m/06xzh</id>
<text>sleep</text>
</freebase>
</task>
```

Output in Ranked Order:

1 put a newborn baby to sleep
2 set a newborn baby to sleep
3 place a newborn baby to sleep
4 pose a newborn baby to sleep
5 position a newborn baby to sleep

6 lay a newborn baby to sleep
7 move a newborn baby to sleep
8 displace a newborn baby to sleep
9 insert a newborn baby to sleep
10 enclose a newborn baby to sleep
11 inclose a newborn baby to sleep
12 stick_in a newborn baby to sleep
13 put_in a newborn baby to sleep
14 introduce a newborn baby to sleep
15 docket a newborn baby to sleep
16 cock a newborn baby to sleep
17 postpose a newborn baby to sleep
18 prepose a newborn baby to sleep
19 step a newborn baby to sleep
20 replace a newborn baby to sleep

4. Results

Run 1

Our STREC@20 score for Run One performed very well with a recall of 81.96%. Our alpha-nDCG@20 was 50.22%. We have just received our results and are still performing our error analysis. Further discusion will be included in our final paper.

Run 2

Our STREC@20 score for one run performed very well with a recall of 76.97%. Our alpha-nDCG@20 was 62.43%. We have just received our results and are still performing our error analysis. Further discusion will be included in our final paper.

Previous Year

The metrics reported last year used nERR-IA@20 and alpha–nDCG@20. We do not have scores for this year yet so we will compare to 2016. The best scores reported for 2016 for nERR-IA@20 were 0.57 and ours was 0.59. The best scores reported for 2016 for alpha–nDCG@20 were 0.70 and ours was 0.62.

5. Conclusions

In reviewing output for Run 2, our team found many of our generated results were either identical to the original or with little to no change in meaning. In understanding the overall task, there is more to be done in terms of finding synsets for query words and phrases.

Future Work

Our intial thoughts for future work is to develop a synthesized solution utilizing ideas from both Run 1 and Run 2. Potentially, we could run our keywords and phrases generated in Run 2 through our Google search algorithm from Run 1.

References

Bennett, Paul N. and White, Ryen W. 2016. Mining Tasks from the Web Anchor Text Graph: MSR Notebook Paper for the TREC 2015 Tasks Track.

Finlayson, Mark Alan. 2014. Java Libraries for Accessing the Princeton Wordnet: Comparison and Evaluation. In H. Orav, C. Fellbaum, & P. Vossen (Eds.), Proceedings of the 7th International Global WordNet Conference. (pp. 78-85). Tartu, Estonia.

Garigliotti, Dario and Balog K. 2016. The University of Stavanger at the TREC 2016 Tasks Track (TREC 2016).

Hagan, M., Kiesel J., Adineh P., Masoud A., Fatehifar E., Bahrami A., Fichtl P., Stein B. 2016. Webis at TREC 2016: Tasks, Total Recall, and Open Search Tracks.

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

Small, Sharon and Tomek Strzalkowski. *2004. HITIQA: A Data Driven Approach to Interactive Analytical Question Answering.* Proceedings of Human Language Technology Conference. Boston, Massachusetts.

Toutanova, Kristina and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000).

Voorhees, Ellen M. 2007. Overview of TREC 2007. In Proceedings of The Sixteenth Text Retrieval Conference (TREC 2007).