A Ranking-based Approach for Multiple-document Information Extraction

Araly Barrera
University of Houston
abarrera7@uh.edu

Rakesh Verma
University of Houston
rmverma@cs.uh.edu

Abstract

This paper presents a ranking-based approach used in a participating system for the TAC 2010 Information Extraction Task. We introduce a prioritization hierarchy consisting of four levels that are used to determine the most important sentences for extraction. The presence of named entities and the document date play major roles in our approach.

1 Introduction

Question-answering based on natural language information is one of the most challenging tasks confronting natural language researchers in the information-driven world of today. Much of the work in this area has been propelled by the need to condense loads of information (including news articles) into shorter indicative or informative summaries for everyday readers. Even though our focus has been on single-document extractive summarization, the UH team decided to participate in the TAC 2010 multi-document information extraction task, consisting in the development of 100-word query-focused summaries from given sets of newswire articles along with an update task. For the various article topics provided in each set, summaries were to be *query-based* by answering a set of aspects that were relevant to that category.

Our system handles this query-based summarization challenge by extracting sentences based on the ranking and prioritization within four different levels:

1. **Level 1:** A sentence's distinct types of entities count.
2. **Level 2:** An article-level rank based on article date.
3. **Level 3:** A normalized score based on a sentence's total entity count.
4. **Level 4:** A sentence-level rank based on our single-document summarization research.

In addition, the information extraction task for TAC 2010 was divided into two parts, A and B. Part A required a 100-word summary that would answer all aspects found for a categorized topic set composed of ten articles. In the case of an "Attacks" categorized topic set, its summary was to provide answers to aspects such as *what happened*, *date*, *location*, *casualties*, *damages*, *perpetrators*, and *rescue efforts*. Part B, otherwise known as the *update task*, also required a 100-word summary from a subsequent set of ten articles but this with the assumption of previous knowledge of part A.

In this paper, we present our system by describing the preprocessing steps involved for both parts A and B as well as insights behind these leveled ranking schemes.

2 Preprocessing Steps

TAC 2010's information extraction task involved multiple-document extraction of news article sets categorized in 45 different topics. The collection of articles for each category was preprocessed by our system before the ranking stage.

The approaches taken for parts A and B were similar for the most part. Both involved the concatenation of all articles within a topic set into a single file and then processing this file for sentence extraction. The extraction procedure for both also revolved around the ranking scheme presented next in this paper. However, because the *update task* of part B required a summary with the special assumption of previous knowledge of the topic, our preprocessing approach for that task involved some extra computation.

In the case of part A, the designated ten topic articles were concatenated before extraction. For part B, all twenty articles (ten from part A plus the ten from part B) for a topic set were collected and then followed by a sentence redundancy removal procedure. Our approach for sentence redundancy removal consisted in removing sentences only taken from the ten articles designated for part B. A simple comparison made from each sentence in part B articles to those in part A would determine rejection. Those sentences containing a 50 percent match or higher to a part A sentence would therefore be removed before proceeding to the sentence extraction stage. Finally, quote elimination was an additional step performed to both sets of articles in an effort to reduce unnecessary quoted-sentence presence in a final summary. This was accomplished by basic quote-character search in a sentence (Ex. "It's like a prison in there," said Jessica Miller).

3 Sentence Extraction

The extraction approach consisted on sentence ranking and tie-breaking. Four different rankings were gathered at each of the four levels mentioned above as we consider the priorities of these levels going from highest (Level 1) to lowest (Level 4). That is, those sentences ranked highest within Level 1 are considered more important for summary extraction than those in Level 2. If for instance a tie in rank was found within Level 1, then the Level 2 ranks for those would break the tie and so on. Ties at any level would involve tie-breaking within the next lower level in this hierarchy.

The following describes the ranking methods used in each level when summarizing an individual topic set.

3.1 Level 1

Prior observations in the given data led us to believe that more the types of named entities a sentence contains, the stronger the likelihood the sentence's capabilities are in answering a set of topic questions such as *What happened?*, *Who was involved?*, and *Where did this happen?*. By named entities, we refer to the objects for which proper nouns are used in a sentence such as "John Doe", considered a *named person*, "U.S", considered a *named location*, and "Federal Aviation Authority", considered a type of *named organization.* To illustrate the power of named entities, consider the following sentence taken from document AFP_ENG_20041103.0679:

> "Theo van Gogh, who had received threats over a controversial film he made about Islam, was shot and stabbed to death Tuesday while cycling on an Amsterdam street."

This sentence alone manages to answer various questions: the "who" (Theo van Gogh, a *named person*), the "what" (a murder over a controversial film about Islam, a *named religious organization*), the "when" (Tuesday, a *named date*), and the "where" (Amsterdam, a *named location*).

Because of the importance of named entities, we used *Jigsaw* [4], an interactive document analyzer that includes a named entity tagger application, as a means of identifying all sentence entities in our experiments to prioritize those containing the most entity types. Five basic named entities are identified: person, location, date, organization, and money.

Level 1 therefore consisted in the rankings of sentences by *distinct* named entity *type* count and is described as follows:

$$Level1Rank(S_i) = TotalDistinctEntityCount(S_i) \tag{1}$$

where function *TotalDistintEntityCount(S_i)* represents the total number of distinct types of entities in sentence S_i.

The following sample sentence, S_0, will be used throughout this paper to illustrate the ranking methods:

> S_0 ="John had a friend Bob who had lunch with John on a Saturday afternoon in Seattle."

There are five named entities mentioned here (John<person>, Bob<person>, John<person>, Saturday<date>, Seattle<location>) but only three unique named entity types (person, date, location). The Level 1 rank for this sentence would be 3.

Those sentences containing all five entity types were thus the highest ranked for extraction selection.

3.2 Level 2

Another major basis of sentence extraction revolved about an article's date, which determines a sentence's relevance to the entire article set's topic. Again, from prior observations, we find that sentences in news articles that are close to the midpoint between the oldest dated and latest article versions are the best communications on a topic. We argue that older articles are relatively bad candidates for summaries due to possibilities of relating incorrect or imprecise versions of when a story freshly broke out and that the latest articles many times provide information lacking a sufficient background review of the topic.

For each of the 45 topic sets containing 10 news articles for tasks A and B, a document-level rank was given to the individual articles based on their date. Once article dates were sorted from earliest to latest (order of 1 to 10), the article ordered midpoint at 5 was ranked first for higher priority. In pyramid form, articles ordered 1 to 4 were ranked 1 to 4 and 6 to 10 were ranked 4 to 1. Ties in date rank would consist in duplicating those ranks. The ranking of sentence S_i is as follows:

$$Level2Rank(S_i) = DateRank(S_i) \tag{2}$$

where function $DateRank(S_i)$ represents the rank given to sentence S_i based on the date scheme mentioned above.

Say our sample sentence, S_0, originates from an article, A, whose date lies within range [October 1 to November 30, 1999], where the oldest article within this topic set is October 1 and the latest is November 30. All articles are sorted from oldest to latest and assume the total number of articles is ten. If A is found to be the fifth latest article in this set, then A would be the highest ranked article with a score of five. Level 2 Rank for $S_0 = DateRank(S_0) = A$'s score = 5.

For Level 1 tie-breaking, Level 2 ranks were then used to prioritize a sentence for extraction.

3.3 Level 3

To fully exploit the named entity idea introduced for Level 1 ranks, this next level consisted in ranking sentences based on a normalized rank score for non-distinct named entity counts. Although this may seem very similar to the ranking basis used in Level 1, we now wish to count the total number of named entities, regardless of their type and regardless of any duplication. The idea behind this approach is similarly based on the potential of named entities in a sentence to provide specific information. Our assumption is that more the number of named entities a sentence contains, the higher the likelihood of answering various questions about a topic. Ranking of sentence S_i for Level 3 is as follows:

$$Level3Rank(S_i) = \frac{TotalEntityCount(S_i)}{|S_i|} \tag{3}$$

where function *TotalEntityCount(S_i)* represents total named entity count found in sentence S_i and $|S_i|$ is the size of the sentence. For the size of a sentence, we did not have time for experimenting with stop word removal, so it consisted of all words. Because sample sentence S_0 contains a total of five named entities (disregarding type, we have: person, person, person, date, location) the Level 2 rank for that sentence would be 5. In the case of a rank tie in Level 2, Level 3 ranks would then be computed to break the tie.

3.4 Level 4

The last level of the presented ranking scheme is based on our prior research on single document summarization. Here, we use sentence scores generated by a system called, SynSem [1]. SynSem fuses syntactic, semantic, and statistical methodologies for individual documents. A *TotalScore* for a sentence S_i is based on three different factors and is a weighted combination of these factors as follows:

$$TotalScore(S_i) = w_1 Position(S_i) + w_2 WordNet(S_i) + w_3 TextRank(S_i)$$

where function *Position* is a score based on the distance a sentence is from a heading, function *WordNet* utilized the WordNet [2] tool to determine semantic relations of a sentence to priority words within a document, and the *TextRank* function utilizes the TextRank algorithm [3] to determine the total popularity score of a sentence's words. The weights we used were obtained by optimizing SynSem using the DUC 2002 collection. Therefore, the ranking of sentence S_i for Level 4 is as follows:

$$Level4Rank(S_i) = TotalScore(S_i) \tag{4}$$

where *Total Score(S_i)* refers to the sentence score produced by our SynSem system. In the case of a tie found in Level 3, the SynSem system and the highest Total Score for those sentences would be used to break the tie. Level 4 rank for S_0 would depend on the SynSem *Total Score.* If a tie with a sentence has carried on when reaching Level 3, for instance, the highest *Total Score* of the two would determine top rank of 1. This being at the lowest level of the hierarchy, the case of a tie in this level would be broken simply by taking the original order of the sentence as a last resort.

Due to SynSem's relative performance over the baselines and the systems that participated in DUC 2002 competition, we felt it equally suitable to use this as one of the ranking level methods when determining the most important sentences for multiple-document extraction.

4 Results and Discussion

An impressive number of runs were submitted to TAC 2010: 41, by 21 participants (up to two per participant) and NIST created two baselines. The method outlined above performed close to the 50 percentile mark for linguistic quality and beat one of the baselines consistently on this measure for both tasks. Linguistic quality included five factors: grammaticality, non-redundancy, referential quality, focus, and structure and coherence. However, we came up short (equal or better than about 20 percent of the runs) on the overall responsiveness score on the A set of articles. On the B set, the update task, our relative responsiveness improved significantly (equal or better than almost 35 percent of the runs).

These results suggest that sentences with more types of named entities and total entities give the summary a better linguistic quality, which could be because of fewer referential issues and higher understandability. The increase in the responsiveness score for the update task suggests that our redundancy removal method is worthy of further investigation. The lower scores for overall responsiveness imply that our scoring method needs improvement.

5 Conclusions

Since much of our focus has been on single-document summarization, these systems were implemented in a very short amount of time leaving us with little time for any refinements and improvements before the submission of runs. Since the submission in 2010, however, we have come up with new ideas for improving our initial system and we are currently in the process of evaluating these improvements. One of ideas we are experimenting with is the identification and deletion of low-yield content from an article. Also, we have observed that the SynSem level, Level 4, was rarely reached in our hierarchical ranking method, so in future, it may be worthwhile to investigate elimination of the current Level 3 tiebreaking method or reversal of Levels 4 and 3. Another possibility is to compare the hierarchical approach with a weighted linear combination.

References

[1] A. Barrera and R. Verma. Automated extractive single-document summarization: Beating the baselines with a new approach. *To appear in ACM's Symposium on Applied Computing*, March 2011.

[2] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, 1998.

[3] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* (EMNLP, 2004), March 2004.

[4] J. Stasko, G. Carsten, and L. Zhicheng. Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization*, 7, No. 2:118–132, 2008.