University of Washington at TAC 2011

Prescott Klassen
University of Washington
Department of Linguistics
Seattle, WA
klassp@uw.edu

1 Introduction

This is the first year the University of Washington Department of Linguistics has participated in the Guided Summarization Track at the Text Analysis Conference. The goal was to implement a basic extractive summarization module that will provide sentence selection and ranking for a more comprehensive future semi-abstractive solution. Due to time constraints, additional components of the overall semi-abstractive solution were unable to be realized in this year's submission. The update task was not generated in respect to the initial summary and aspects were not explicitly addressed in our system.

2 System Description

The module integrates the Stanford CoreNLP[1] package to identify features and implements Log Likelihood Ratio (LLR) [Dunning, 1993] to calculate signature terms [Lin and Hovy, 2000]. Sentences are ranked by their cumulative signature feature LLR values. The 'clean data' version of TAC 2010 and 2011 documents was used for training and testing.

2.1 Program Flow

The initial and update document sets are converted from the 'clean data' format to text files of line-delimited sentences in document order. Only sentences with value of 1 from the 'clean data' files are included in the output files. The Stanford coreNLP package is used to annotate sentences which are output to a stand-off XML representation.

[1] http://nlp.stanford.edu/software/corenlp.shtml

Features are extracted from the XML annotation files and output into line-delimited feature vector/sentence pairs. Feature vectors are populated with feature/value pairs where values are counts of the feature within the document.

The Log Likelihood Ratio (LLR) is calculated for topic features and line-delimited feature/value pairs are output to a file. LLR values for each sentence are calculated and then each sentence is filtered for length, noise, and redundancy using sentence minimum and maximum sizes, regular expressions to filter noisy strings for the sentence, and a rejection of any sentence that shares more than 70% of its tokens with previously filtered sentences. Sentences are ranked by the cumulative total of their features' LLR value and selected for output to the final summary. Selection of sentences continues until the 100 token length limit is met or the total is within 7 tokens.

2.2 Architecture

Developed primarily in Java, the system integrates StanfordCoreNLP annotators to provide stand-off annotation in XML. The Stanford coreNLP annotators used are: tokenize(tokenization), ssplit (sentence splitting), pos(parts-of-speech tagging), lemma (morphological analysis), ner (named entity recognition), parse (syntactic parsing), and dcoref (coreference resolution).

2.3 Features

RUN ID 8 features are realized as super-tags composed of lemma + part-of-speech (Sheriff_NNP John_NNP) and named entity + lemma

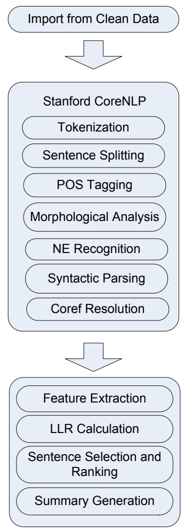

Figure 1: System Diagram

(Tuesday_NNP Tuesday_DATE afternoon_NN afternoon_TIME). Co-references are resolved and output as the original reference. Parts-of-speech are constrained to only nouns, verbs, adjectives, and prepositions. Named Entities include PERSON, ORGANIZATION, LOCATION, DATE, MONEY, and MISC.

RUN ID 29 features are realized as supertags composed of lemma + part-of-speech + dependency information. Parts-of-speech features are restricted to verbs, nouns, and adjectives (look_VERB_dep_border_NOUN). Dependency triples are also included as features (come_VERB_nsubj_people_NOUN). Named Entities and resolved co-references are included in the output.

2.4 Log Likelihood Ratio

The Log likelihood Ratio implemented in the solution is from [Dunning, 1993].

$$\lambda = \frac{max_p L(p, k_1, n_1) L(p, k_2, n_2)}{max_{p1,p2} L(p_1, k_1, n_1) L(p_2, k_2, n_2)} \tag{1}$$

where

$$L(p, k, n) = p^k (1-p)^{n-k} \tag{2}$$

and $p_1 = \frac{k_1}{n_1}, p_2 = \frac{k_2}{n_2}, p = \frac{k_1+k_2}{n_1+n_2}$

k_1 is the count of the feature within the summary document set, n_1 is the total number of features in the summary document set, k_2 is the count of the feature within the entire corpus with the exclusion of the summary documents, and n_2 is the total number of features within the entire corpus excluding the total number of features in the summary documents.

The logarithm of the likelihood ratio used in the LLR calculation component of the implemented system is:

$$\begin{aligned} -2log\lambda = 2[&logL(p_1, k_1, n_1) + logL(p_2, k_2, n_2) \\ &-logL(p, k_1, n_1) - logL(p, k_2, n_2)] \end{aligned} \tag{3}$$

3 Evaluation

Two runs submitted by U of W were ID 8 and ID 29. The ROUGE scores for the two runs were low. In most cases, the F-measure scores for both runs were in between those of baseline ID 1 and Baseline ID 2. ID 29 was the better performing of the two runs, although the difference between the two runs was small. ID 29 used dependency features for LLR signature terms rather than the simple parts-of-speech features in ID 8.

3.1 ROUGE Results for ID 8 and ID 29

Table 1: ID 8 Summary A

ROUGE	Recall	Precision	F-Measure
1	0.3133	0.3077	0.3104
2	0.0598	0.0584	0.0591
3	0.0206	0.0201	0.0203
4	0.0101	0.0099	0.0100
L	0.2648	0.2602	0.2624
W-1.2	0.0938	0.1654	0.1196
SU4	0.0979	0.0961	0.0970

Given the historically strong results of previous DUC and TAC LLR implementations [Nenkova and McKeown, 2011], the ROUGE scores of runs ID 8 and ID 29 were unexpected. In Figures 1-4, the two runs are consistently ranked lower than or close to the baseline runs ID 1 and ID 2. One of the factors in the low ranking of ID 8 and ID 29 might be the pervasive location/date/time information in article first sentences that was not delimited in the origi-

Table 2: ID 29 Summary A

ROUGE	Recall	Precision	F-Measure
1	0.3364	0.3296	0.3328
2	0.0757	0.0741	0.0749
3	0.0286	0.0279	0.0283
4	0.0145	0.0141	0.0143
L	0.2878	0.2821	0.2848
W-1.2	0.1025	0.1802	0.1306
SU4	0.1134	0.1111	0.1122

Table 3: ID 8 Summary B

ROUGE	Recall	Precision	F-Measure
1	0.3074	0.3087	0.3079
2	0.0555	0.0555	0.0554
3	0.0196	0.0196	0.0196
4	0.0108	0.0108	0.0108
L	0.2643	0.2654	0.2648
W-1.2	0.0935	0.1687	0.1202
SU4	0.0969	0.0972	0.0970

Table 4: ID 29 Summary B

ROUGE	Recall	Precision	F-Measure
1	0.3210	0.3187	0.3196
2	0.0690	0.0685	0.0687
3	0.0271	0.0270	0.0270
4	0.0145	0.0145	0.0145
L	0.2770	0.2752	0.2759
W-1.2	0.1000	0.1787	0.1282
SU4	0.1058	0.1051	0.1054

nal clean data files or filtered by the application. The regular expressions written to filter this information from first lines in the training data did not remove the new pattern for this information in the TAC 2010 KBP corpus.

4 Conclusions and Future Work

The extractive module developed for TAC 2011 by the U of W team was intended to provide a baseline LLR extractive summarization module for sentence selection and ranking in a larger more comprehensive semi-abstractive system. Based on the results of both runs, ID 8 and ID 29, features based on dependency structures marginally outperformed simple part-of-speech based features. We will be adding filtering expressions to our extractive module to remove new location/date/time patterns in the TAC 2010 KBP data that were not caught by our original system, which may possibly slightly improve our scores.

Our long term goal is to extract a subset of high value sentences with the extractive module in order to reduce the computational costs and the grammaticality barrier of deep processing tools.

References

Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74, 1993.

Chin-Yew Lin and Eduard Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 495–501, 2000.

Ani Nenkova and Kathleen McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2-3):103–233, 2011.

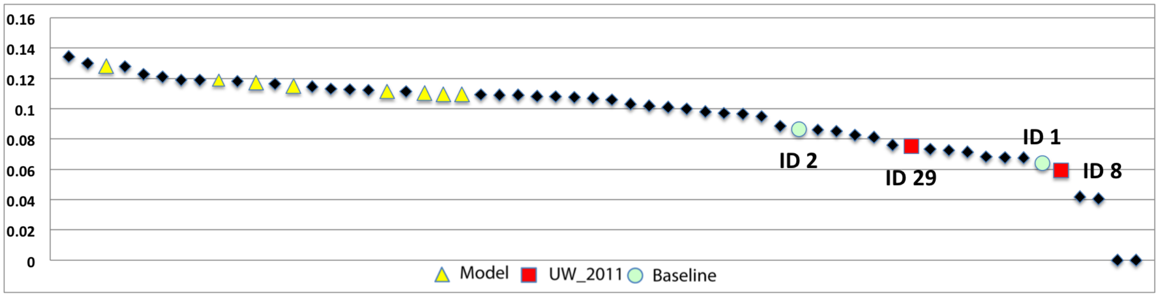

Figure 2: ROUGE 2 Summary A

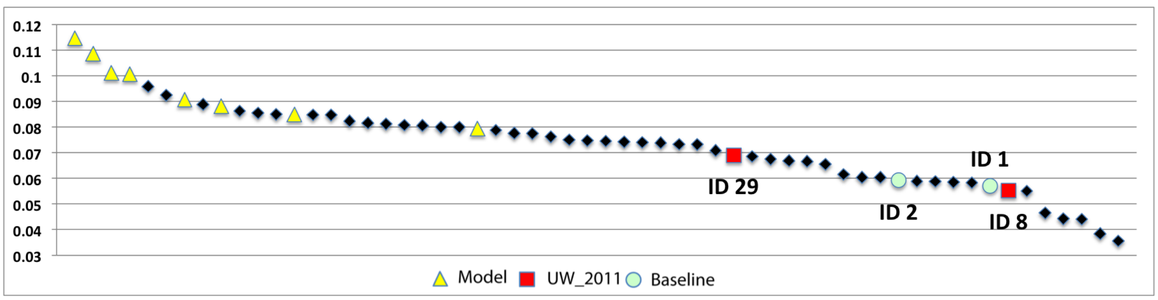

Figure 3: ROUGE 2 Summary B

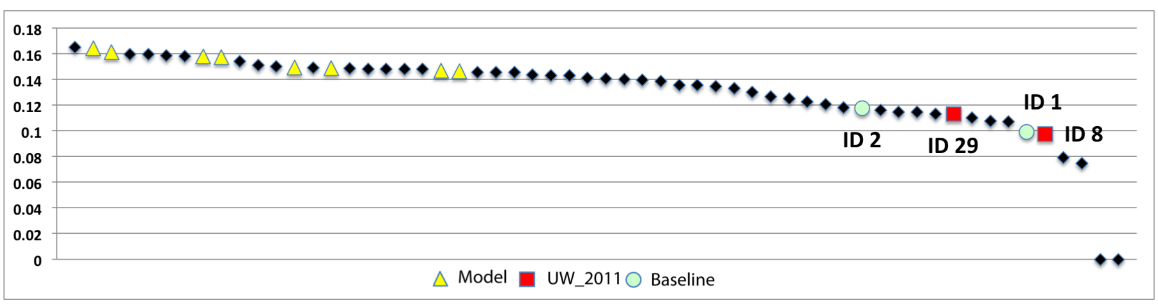

Figure 4: ROUGE SU4 Summary A

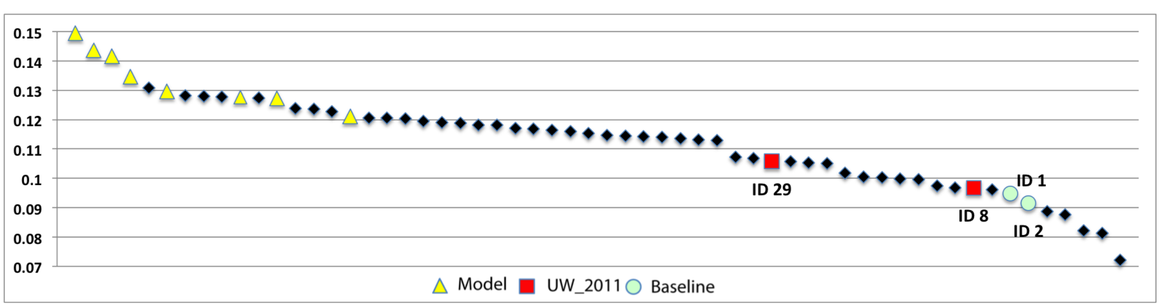

Figure 5: ROUGE SU4 Summary B