Thomson Reuters at TAC 2009: ContextChain and Fractional Conditional Compressibility of Models

Frank Schilder and Ravi Kondadadi and Harsha Veeramachaneni
Thomson Reuters Corporation
Research & Development
610 Opperman Drive
St. Paul, MN 55104, USA
FirstName.LastName@ThomsonReuters.com

Abstract

This paper contains the results for the FastSum system and a simple baseline system for the TAC 2009 main task – update summarization –. For the pilot task of Automatically Evaluating Summaries of Peers (AESOP), we present two novel metrics. The first metric called ContextChain *is an extension of a recently proposed metric AutoSummENG that is based on comparing n-gram graphs of the model summaries and the automatically generated summaries. Our modification of the generated n-gram graphs is based on co-reference chains extracted from the summaries. The n-gram graph is then generated from the context information of these referents.*

*Our second metric called Fractional Conditional Compressibility of Models (*FraCC*) is based on the Burrows-Wheeler compression algorithm. For this evaluation metric, we use an estimate of the conditional "compressibility" of the model summaries given the system summary. The conditional compressibility is defined as the increase in the compressibility of the model summary when the system summary has been observed.*

In addition to presenting our two new approaches to automatically evaluating summaries, we will introduce two new evaluation measures for automatic metrics called Correlation Recall and Correlation Precision and discuss how they can cast more light on the coverage and the correctness of the evaluation metrics for summarization.

1. Introduction

This paper reports on the results we received for our FastSum system [15] as well as a simple baseline of first sentences for the update summarization task. Although we did not carry out any modifications for our FastSum system, we received competitive results for both sub-tasks of the update summarization task. The simple baseline which collects the first sentences of the most recent articles also shows a stronger performance than the NIST baseline 1 that only extract the leading sentences of the most recent document for task A (i.e., multi-document summarization).

For the pilot task of Automatically Evaluating Summaries of Peers (AESOP), we present two novel ways of automatically evaluating summaries. The first metric called ContextChain [16] is an extension of a recently proposed metric AutoSummENG [5] that is based on comparing n-gram graphs of the model summaries and the automatically generated summaries. Our modification of the generated n-gram graphs is based on co-reference chains extracted from the summaries. The n-gram graph is then generated from the context information of these referents. Our second metric called FraCC is based on the Burrows-Wheeler compression algorithm. For this evaluation metric, we use an estimate of the conditional "compressibility" of the model summaries given the system summary. The conditional compressibility is defined as the increase in the compressibility of the model summary when the system summary has been observed.

Both metrics showed high correlation values when compared with the two manual metrics used (i.e., pyramid and responsiveness). The highest Pearson correlations we received for task B, for example, was 0.937 and 0.946 for ContextChain and FraCC, respectively. FraCC also was able to distinguish between different systems very accurately, as shown by the discriminative power analysis carried out by NIST. FraCC was among the top 7 performing metrics resulting from the ANOVA carried out by NIST on the scores produced by each metric.

In addition to presenting our two new approaches to automatically evaluating summaries, we introduce two new Evaluation measures for automatic metrics called Correla-

tion Recall and Correlation Precision and discuss how they can cast more light on the coverage and the correctness of the respective metric.

Correlation Recall is computed for the top *n* performing systems only and not the entire set of systems. The two score vectors for the automatic and the manual metric are sorted according to the manual metric. This could mean that systems that obtained high scores from the automatic metric, but low manual scores were not considered for the correlation evaluation. The resulting coefficient numbers indicate how well the top systems according to the manual metric are covered by the automatic metric.

Correlation Precision, on the other hand, is computed for the top *n* systems sorted by the automatic metric. This could mean that systems that obtained high manual scores, but low automatic metric scores were not considered for the correlation evaluation. The Correlation Precision numbers seem to be a better indicator for how good the automatic metric can predict strong performing systems.

2. Related Work

ROUGE [12, 13] is one of the first automatic summarization evaluation metrics proposed. ROUGE uses lexical n-grams to compare human written model summaries with automatically generated summaries. Hovy et. al. Later, [7] proposed an approach to automatic evaluation based on the concept of Basic Element. A Basic Element (BE) is a semantic unit extracted from a sentence such as subject-object relation, modifier-object relation. Systems with higher overlap of system-summary BEs and human-summary BEs get higher BE scores.

Recently, AutoSummENG was introduced as a summarization evaluation method that evaluates summaries by extracting and comparing *graphs* of character and word n-grams [5]. Both the model and system summaries are represented as graphs. Edges in the graph are created based on the adjacency relation between n-grams. The edges are weighted according to the distance between the neighbors or the number of co-occurences with in the text. Similarity between two graphs is computed as number of common edges. Similarity can also include the weights of the common edges.

Barzilay and Lapata (2008) [1] describes a framework to capture local textual coherence. Their approach is based on the assumption that the distribution of discourse entities in a coherent text shows certain regularities as formalized in Centering theory [6]. The entity distribution is represented as a grid using coreference relations, syntactic knowledge and salience. They used a ranking SVM to model the regularities in a coherent text. They demonstrated good performance on text ordering, summarization evaluation and readability assessment. Our approach to summarization evaluation uses similar links between named entities and definite description. We, however, use these links to generate an n-gram graph, as proposed by [5]. The context chains differ from the n-grams generated for the AutoSumm metric by the type of links generated.

Two other proposals for new evaluation metrics address the question of improving the evalution metric in general, but they do not address the problem of low correlations for top n system discussed by this paper. Tratz and Hovy (2008) [18] describe a new implementation of the BE method, called BE with Transformations for Evaluation (BEwTE) that includes a significantly improved matching capability using a variety of operations to transform and match BEs in various ways. BEwTE uses a set of transformations to match BEs that are semantically similar but are lexically different. The transformations include mapping adverbs to adjectives and vice versa, dropping periods in abbreviations, expanding or shortening names, matching WordNet synonyms. The score of a BE is computed as a linear combination of similarities of the BE with the matching reference summary BEs. Louis and Nenkova (2008) [14] use features based on distribution of terms in the input summary and the model summary. They use KL, JS Divergence and cosine similarity to compute the similarity of term distribution of the input and the model summary. The features are based on distributional similarity, summary likelihood and topic signatures. Features include KL divergence, JS divergence and cosine similarity between the input and the model summary, percentage of summary composed of topic signatures from input. They also evaluated a feature that is a linear combination of all other features using linear regression.

3 Update summarization

This year's update summarization task was the same as the last year's task with two changes. Firstly, NIST assessors selected topics that were closer together in time than last year's topic. Secondly, the overall responsiveness score was evaluated on a 10-point scale rather than a 5-point scale.

3.1 Task description

The update summarization task is defined as follows. Given a topic, the task is to write 2 summaries (one for Document Set A and one for Document Set B) that address the information need expressed in the corresponding topic statement:

1. The summary for Document Set A should be a straightforward query-focused summary.
2. The update summary for Document Set B is also query-focused but should be written under the assump-

```
<title>
Apple Computer switch to Intel chips
</title>
<narrative>
Trace plans for and progress of the
switch to Intel chips by Apple Computer.
</narrative>
```

Figure 1. A TAC 2009 sample topic

tion that the user of the summary has already read the documents in Document Set A.

Automatic summarization systems need to produce summaries that are well-organized using complete sentences. the limit for is summary is 100 words (whitespace-delimited tokens). Within a topic, the document set A must be processed before document set B. A sample topic can be found in Figure 1.

NIST used three different baselines this year:

1. Baseline 1 returns all the leading sentences in the most recent document until summary limit of 100 words is reached.
2. Baseline 2 is based in a copy of one of the model summaries for the document set, but with the sentences randomly ordered. This baseline was meant to determine the effect on the linguistic quality, but received artificially high scores from the automatic metrics because the model summary that the baseline is generated from was not excluded.
3. Baseline 3 produces summaries that consist of sentences that have been manually extracted from the respective document set. This Hex-Tac baseline (Human EXtraction for TAC) was contributed by a team of five human'sentence-extractors' from the University of Montreal.

3.2 FastSum and a simple baseline

For this year's update summarization task, we submitted two runs. The first run was produce by our FastSum system trained on previous years' data. In addition, we designed a simple baseline system called first line baseline.

3.2.1 FastSum

FastSum is a multi-document summarization system that uses a regression SVM for training a sentence classifier for good summary sentences similar to [11]. A part of FastSum is a filtering component that sorts out sentences that are unlikely to be in a good summary (e.g., no word overlap between query and sentence, difference in length).

Pre-processing and filtering The pre-processing module carries out tokenization and sentence splitting. In addition, a sentence simplification component based on a few regular expressions removes unimportant components of a sentence (e.g., *As a matter of fact,*). This processing step does not involve any syntactic parsing. As an initial filter, we ignore all sentences that do not have at least two exact word matches or at least three fuzzy matches with the topic description.[1]

Feature set Features are mainly based on frequencies of words in sentences, documents and document clusters. The features we used can be divided into two sets: word-based and sentence-based. Word-based features are computed based on the relative frequency of words for different segments (i.e., cluster, document, topic title and description). At runtime, the different relative frequencies of all words in a candidate sentence, s, are added up and normalized by the length $|s|$. Sentence-based features include the length and position of the sentence in the document.

Training In order to learn the feature weights, we trained a regression SVM [8] on the previous year's news data using the same feature set. In regression, the task is to estimate the functional dependence of a dependent variable on a set of independent variables. In our case, the goal is to estimate the "summary-worthiness" of a sentence based on the given feature set. In order to get training data, we computed the word overlap between the sentences from the document clusters and the sentences in TAC 2008 model summaries. We associated the word overlap score to the corresponding sentence to generate the regression data.

3.2.2 1st line Baseline

This baseline selects the temporally ordered first sentences from each article until the word limit is reached. A sentence is not added if the cosine similarity between the sentence and the summary is more than 0.7.

3.3 Evaluation

Our FastSum system received competitive results again, although not as high as for TAC 2008. FastSum received high Pyramid scores and linguistic quality scores for task A, as indicated by Figure 1. The system also achieved high scores for Task B with respect to responsiveness and linguistic quality, but not for the Pyramid score (cf., Figure 2). The first line baseline, on the other hand, received mediocre scores for most of the metrics except for the automatic metrics for task B. For ROUGE-2 and BE, the first line baseline

[1] Fuzzy matches are defined by the OVERLAP similarity [2] of at least 0.1.

System	Responsiveness	Pyramid	Ling. Quality	ROUGE-2	BE
Best System	5.159	6.5	5.932	0.12184	0.06379
Baseline 1	3.636	3.182	6.705	0.06315	0.02916
Baseline 2	6.364	11.977	5.477	0.33133	0.2483
Baseline 3	6.341	6	7.477	0.10633	0.05333
FastSum	4.455	5.295	5.545	0.09366	0.04382
Rank	13	9	7	21	27
1st line Baseline	4.205	4.182	4.795	0.09307	0.04781
Rank	20	27	23	22	21

Table 1. TAC 2009 update summarization results: task A

got top scores sometimes even higher than some of the human summarizers, as shown in Figure 2.

Our FastSum system normally received higher results than the first line baseline, but for the automatic scores for Task B this order was clearly reversed. This seems to indicate that comparing systems' performances solely on these two automatic metrics is not very reliable and may lead to wrong conclusions. It may also point to problems with evaluating systems for Task B. This task may be substantially different from the multi-document summarization task and previously proposed metrics may not be able to capture the essentials of this task.

The following section contains a more in-depth discussion on how the current summarization tasks should be evaluated.

4. Pilot task: AESOP

This year's pilot task was called the Automatically Evaluating Summaries of Peers (AESOP) task. The purpose of this task was to encourage the development of systems that automatically evaluate the quality of summaries. Participants were able to run their automatic metrics on systems' and human summaries from the TAC 2009 Update Summarization task and results were evaluated by NIST in terms of correlation to two manual metrics: the (Modified) Pyramid score, which measures summary content, and Overall Responsiveness, which measures a combination of content and linguistic quality.

4.1 Task description

The AESOP task was carried out with the 2009 Update Summarization task. NIST supplied all automatically generated summaries and document sets for 44 topic statements as well as four human-authored summaries for each topic set.

A system submitted for this task had to produce two sets of numeric summary-level scores:

All Peers case : a numeric score for each peer summary, including the model summaries.

No Models case : a numeric score for each peer summary, excluding the model summaries.

The **All Peers** score should be helpful for differentiating between human vs automatic summarizers, whereas the **No Models** score focuses on how well an automatic metric can evaluate automatic summaries.

4.2 Two new metrics

4.2.1 Context Chains and n-gram graphs

[5] proposed a method called AutoSummENG that generates and compares n-gram graphs for the model summaries and the automatically generated summaries to evaluate the quality of automatic summaries.

The AutoSummENG summarization evaluation metric is based on the similarity between the n-gram graph representations for the generated system summaries and model summaries. An n-gram graph can be generated for word or character windows. An 2-gram graph for n=2 for the following sentence can constructed by first generating all 2-grams:

A quick brown fox jumps over the lazy dog.

Figure 2 shows the complete graph generated from this sentence. In addition, weights on the edges can indidcate the distance between the neighbors or the number of occurrences in the text. By creating edges between the adjacent n-grams, this approach takes the contextual information into consideration as opposed to approaches that only use the n-gram overlap between the system and model summaries. Similarity between the graphs is computed via the Co-occurrence Similarity, Value Similarity, and the Size Similarity. Co-occurrence Similarity is based on the number of common edges between the graphs. Value Similarity is similar to the Co-occurrence Similarity except that it also includes the weights of the edges. Size Similarity is the ratio of number of edges of the smaller graph to the number

System	Responsiveness	Pyramid	Ling. Quality	ROUGE-2	BE
Best System	5.023	0.307	5.886	0.10417	0.06364
Baseline 1	4.318	0.16	6.455	0.05115	0.02417
Baseline 2	6.182	0.69	5.886	0.31932	0.25042
Baseline 3	6.114	0.329	7.25	0.09799	0.05669
FastSum	4.273	0.21	5.864	0.07586	0.04125
Rank	8	20	2	24	21
1st line Baseline	4.136	0.238	4.909	0.08819	0.05168
Rank	12	13	17	8	7

Table 2. TAC 2009 update summarization results: task B

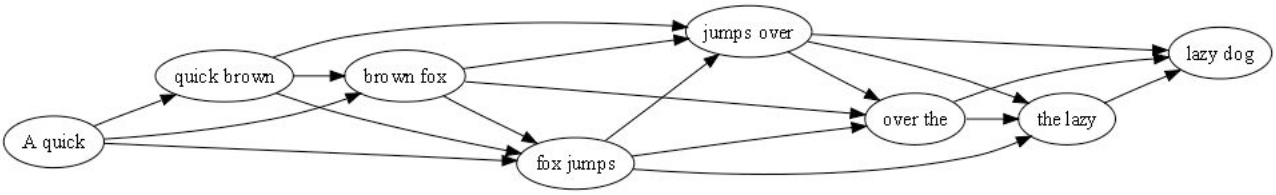

Figure 2. An n-gram graph

of edges in the bigger graph. The overall similarity is computed as a function of these three similarity measures. [2]

They show that their approach is superior over past automatic metrics such as ROUGE and BE for the DUC 2005, 2006 and 2007 summarization tasks in terms of reaching higher Pearson correlation coefficients.

Our approach is an extension of AutoSummENG that generates n-gram graphs based on co-reference chains. Our approach also models local coherence by establishing chains of potentially co-referent named entities and definite descriptions. The n-gram graph is then generated from the context of these referents. Consider the the beginning of a news story shown in figure 3. These n-grams can be seen as the events the entities mentioned in the summaries are involved in and the links determine the sequence in which the events should be mentioned. The links, therefore, capture the local coherence, as found in the model summaries. Note that this is a main difference between our approach and the other purely n-gram based approaches. An automatically generated summary may share lots of n-grams with the model summaries, but the sequence of how the events are presented may be incoherent and hence decreases the readability of the summary.

We implemented our approach within the AutoSumm GUI that is freely available. For the named entity extraction and chunking, we used LingPipe's named entity tagger and chunker.[3]

[2]See [5] on how to compute these scores.

[3]Baldwin, B. and B. Carpenter. LingPipe. `http://www.aliasi.com/lingpipe/`.

4.2.2 Fractional Conditional Compressibility of Models (FraCC)

Our second metric called FraCC is based on the Burrows-Wheeler compression algorithm. For this evaluation metric, we use an estimate of the conditional "compressibility" of the model summary given the system summary. The conditional compressibility is defined as the increase in the compressibility of the model summary when the system summary has been observed. The compressibility of a string is estimated through the move-to-front entropy (also called local entropy) of the Burrows-Wheeler transform of the original string. Since the Burrows-Wheeler transform involves just the construction of a suffix array, the computation of our compression based evaluation metric is linear in time and space in the size of the model and machine summary strings.

In order to judge the similarity of the system summary S, to the model summary M, we propose to use the difference in compressibility of M when S is not seen to when S is given. This metric basically captures the reduction in the uncertainty in M when S is known.

We define the compressibility $C(M)$ of the string M by $c(M) = \frac{H(M)}{|M|}$

and the conditional compressibility of string M over an alphabet A given another string S over the same alphabet as $c(M|S) = \frac{H(S+M)-H(S)}{|M|}$

where $S + M$ is the concatenation of the strings S and M, H(S) is the entropy of string S, and $|M|$ is the length of the string M.

The Justice *Department* is *conducting* an *anti-trust trial* against **Microsoft Corp** with *evidence* that the *company* is *increasingly attempting* to crush competitors. **Microsoft** is *accused* of *trying* to *forcefully buy* into markets...

All context 4-grams (minus stop words) for the named entity Microsoft:

Department_conducting_anti-trust_trial
evidence_company_increasingly_attempting
accused_trying_forcefully_buy

Two context chains are generated:

Department_conducting_anti-trust_trial—accused_trying_forcefully_buy
evidence_company_increasingly_attempting—accused_trying_forcefully_buy

Figure 3. Example text and 2 example context chains generated for one named entity

The fractional conditional compressibility of M given S is then measured by $r(M|S) = \frac{c(M)-c(M|S)}{c(M)}$

We use $r(M|S)$ as the similarity metric to measure the similarity of a system summary S to model summary M.

Note that although similar in principle to the similarity metric proposed in [10], our similarity is asymmetric. In order to compute the similarity metric we need to estimate the the entropy $H(S)$ for a string S, which we define below.

Estimation of string entropy by BWT We use the Move-To-Front (MTF) entropy of the Burrows-Wheeler transform of the given string S as an estimate for its entropy $H(S)$.

The Burrows-Wheeler Transform (BWT) is a permutation on a string over an ordered alphabet, that can be reversed with very little additional information [3]. BWT forms the basis of the bzip2 algorithm, and allows high compression at computational and space complexity linear in the length of the string. It is implemented using the suffix array data structure.

The BWT is a block sorting transform and results in a string that has long runs of symbols, and therefore is compressible by run-length coding. BWT is often followed by the Move-To-Front (MTF) coding. The MTF encoding of the string is performed by traversing the string assigning to each symbol the position number of that symbol in the alphabet and then moving the symbol to the front of the alphabet. Therefore a sequence of repeated symbols will be encoded as zeros for all but the first occurrence.

In [9] the MTF coding is used to define the MTF entropy of a string R as $\text{MTFE}(R) = \sum_i \log(\text{MTF}(R)_i + 1)$, where $\text{MTF}(R)_i$ is the i^{th} symbol of the MTF coding of the string R.

Now we define $H(S)$, the entropy of string S as $H(S) = \text{MTFE}(\text{BWT}(S))$, where $\text{BWT}(S)$ is the BWT of string S.

Some technical details For the implementation of our FraCC similarity score, we used a word level representation of the strings. Our alphabet of symbols contained all the words in two strings being compared. The words were normalized by lower casing and removing punctuation. Because BWT needs an ordered alphabet, we used the lexicographic order on the words in the alphabet.

4.3 Evaluation

NIST provided evaluation results with respect to the following aspects:

1. Correlation with the manual metric. NIST calculated the Pearson's, Spearman's, and Kendall's correlations between the summarizer-level scores produced by each submitted metric and the manual metrics (Overall Responsiveness and Pyramid).

2. Discriminative Power compared with the manual metric. NIST carried out a one-way analysis of variance (ANOVA) on the scores produced by each metric (automatic or manual). The output from ANOVA was submitted to MATLAB's multiple comparison procedure, using Tukey's honestly significant difference criterion.

NIST ran two baselines:

1. ROUGE-SU4 scores, with stemming and keeping stopwords

2. Basic Elements (BE) scores. Summaries were parsed with Minipar, and BE-F was extracted. These BEs were matched using the Head-Modifier criterion.

We will discuss only the correlation measures for the No Models case, because the All Peers case produced very high

correlations for many metrics including ours. The No Models case is also more interesting for the normal use case for an automatic metric where one wants to improve their summarizer. Tables 3 and 4 summarize the different correlation measures, as computed by us by leaving out baselines 2 and 3.[4] Note that the correlation was computed for all baseline systems, too. Baseline 2, however, had an unfair advantage because it incorporated one of the model summaries and the scores are artificially high which may influence the overall correlation measurement. In the following section, we computed correlation coefficients without baseline 2 and 3 because we suspect that these may skew the correlations.

Taking this caveat into account, a first analysis of the correlation scores seems to suggest that the standard automatic metrics correlate well with Pyramid, but not so well with Responsiveness. Our two metrics[5] perform reasonably well and receive similar weights to the two standard automatic metrics ROUGE-SU4 and BE. The "best" metric for most correlation coefficients (but not all) is metric 26 [4] which scored an almost perfect Pearson r value of 0.978 for Task A with the Pyramid metric.

After this first analysis of the correlation coefficient, it may be concluded that we now have multiple automatic metrics that correlate well with the Pyramid scores, but not so well with the Responsiveness scores. We continue our analysis with a more in-depth view of how the different metrics perform for the top *n* systems in the next section.

In addition to the Pearson, Spearman, and Kendall scores, NIST also carried out an ANOVA in order to determine the discriminative power of the metrics. This analysis focuses on the question of whether the metric is able to pick up significant differences between systems. FraCC was always among the seven best performing metrics when the ANOVA was conducted for model and non-model summaries. For the summaries A for the Pyramid metric there were 432 significant differences between summaries which FraCC (and six other metrics) recognized. ROUGS-SU4 and BE-HM – the two baseline metrics – only detected 227 and 97, respectively.

5. Correlation precision and recall

For our follow-up experiments, we first excluded the two baselines 2 and 3 and recomputed the Correlation coefficients for Pyramid and Responsiveness. Interestingly enough, the values for Responsiveness improved for all metrics ranging from 0.860 to 0.91. From conclude from this improvement in the correlation coefficient that baseline 2 was an outlier that skewed the overall results.

[4]Note that NIST included all baselines which resulted in different correlation values.

[5]We report only the best ContextChain variation here. The best variation was achieved by uni-grams graphs.

After removing the two baselines 2 and 3, we ran the TAC 2009 results for each system for different n top systems ($n = 10, 15, 20, 25, 30, 35, 40, 45, 50$). For the obtained results, we computed Pearson coefficients in two ways:

- Responsiveness-sorted: The two vectors of results were sorted according to Responsiveness scores. This could mean that systems that obtained high scores from the automatic metric, but low Responsiveness scores were not considered for the correlation evaluation.
- Automatic evaluation metric-sorted: the two vectors of results were sorted according to the automatic metric. This could mean that systems that obtained high Responsiveness scores, but low automatic metric scores were not considered for the correlation evaluation.

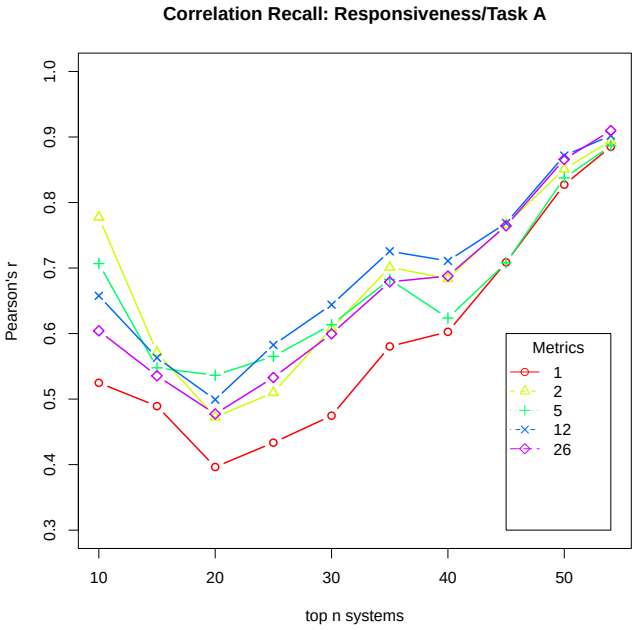

Figure 4. Correlations for n top systems sorted by Responsiveness for Task A

Figure 4 and Figure 5 show the Pearson coefficients for the top 10-54 systems, respectively, if sorted according to Responsiveness. This set-up of the experiment focuses on the top n systems determined by the manual evaluation metric. An automatic metric that shows high coefficients throughout the different number of top systems, shows high coverage (or recall) of the top performing systems. We define this set-up as *Correlation Recall*.

	Pyramid			Responsiveness		
Metric	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall
ROUGE-SU4 (1)	0.921	0.923	0.785	0.767	0.805	0.629
BE (2)	0.857	0.936	0.791	0.692	0.842	0.669
ContextChain (5)	0.899	0.857	0.684	0.769	0.776	0.615
FraCC (12)	0.901	0.947	0.815	0.756	0.849	0.674
"best metric" (26)	0.978	0.942	0.810	0.872	0.847	0.678

Table 3. Different correlation coefficients between automatic metrics and Pyramid and Responsiveness for Task A

	Pyramid			Responsiveness		
Metric	Pearson	Spearman	Kendall	Pearson	Spearman	Kendall
ROUGE-SU4 (1)	0.940	0.863	0.708	0.729	0.719	0.565
BE (2)	0.924	0.932	0.801	0.694	0.816	0.671
ContextChain (5)	0.937	0.880	0.734	0.756	0.722	0.557
FraCC (12)	0.946	0.932	0.781	0.734	0.804	0.657
"best metric" (26)	0.970	0.903	0.768	0.814	0.742	0.607

Table 4. Different correlation coefficients between automatic metrics and Pyramid and Responsiveness for Task B

Conversely, an automatic system that shows a consistently high coefficient for systems sorted according to the automatic metric, is reliable in terms of its precision. In other words, a high automatic score is likely to indicate a high performing system in terms of Responsiveness. We define this set-up as *Correlation Precision*.

The Correlation Precision, on the other hand, seems to be a better indicator for how good the metric can predict strong performing systems.

Figure 6 contains the Pearson coefficients for the top n systems for all metrics shown in tables 3-5 for Task A. The coefficient values vary a lot for the top n systems and in particular BE coefficient values are generally low and only high n allow the conclusion that the metric correlates with the human evaluation metric. ROUGE-SU4 shows high correlations for small n, but correlation values dip below 0.5 for n = *15, 20*. Our metrics stay above 0.5 for all n, whereas the "best" metric shows lower correlation values for n=*15, 20*.

On the contrary, all metrics show high correlation values for Task B, as indicated by Figure 7. Especially BE shows very high coefficient values for the top 10, 15 and 20 systems.

6. Conclusions

We reported competitive results for our FastSum system for the Update summarization tasks and we introduced a new baseline that showed reasonable good results. FastSum showed high Responsiveness scores for Task B and in particular good linguistic quality scores again. The later score proves again the usefulness of the first-sentence classifier we developed last year [17]. The first line baseline is motivated by the same observation that news messages start often with good summary sentences. This strategy plays well for Task B – the update summarization, as the higher scores for this baseline indicate.

The two automatic metrics we proposed – ContextChain and FraCC – received high correlation scores for the different tasks. In particular, FraCC was among the top 7 metrics that posses a high discriminative power for Pyramid and Responsiveness.

We carried out a more in-depth analysis of the correlation between the automatic metrics and the manual metrics Responsiveness and Pyramid. We were able to show that some metrics show lower or no correlations for the top n systems compared to the entire set of 54 systems.

This analysis allows for a better differentiation of metrics with respect to their correlation for the top n systems. However, one may not conclude from this analysis that certain automatic metrics that performed poorly for top n systems this year will show a similar low correlation next year, for example. Given the small sample size, the confidence intervals are relatively large compared to the full set of systems. We can, however, determine which metrics show low or now correlations for top n systems for this year's results.

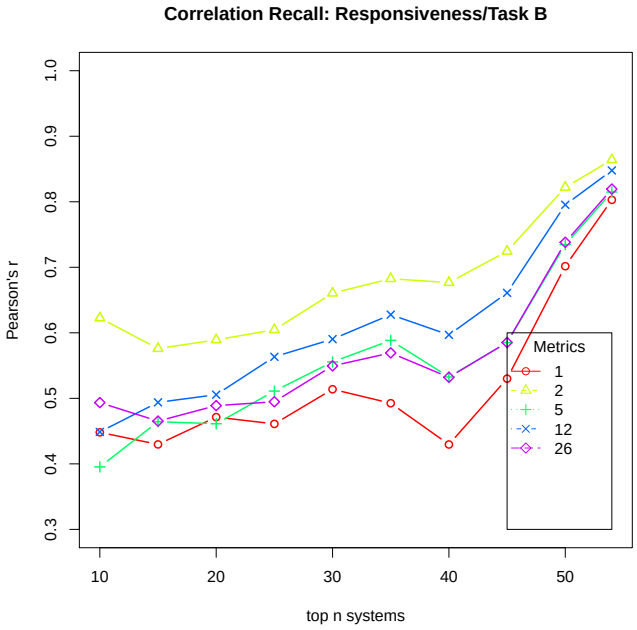

Figure 5. Correlations for n top systems sorted by Responsiveness for Task B

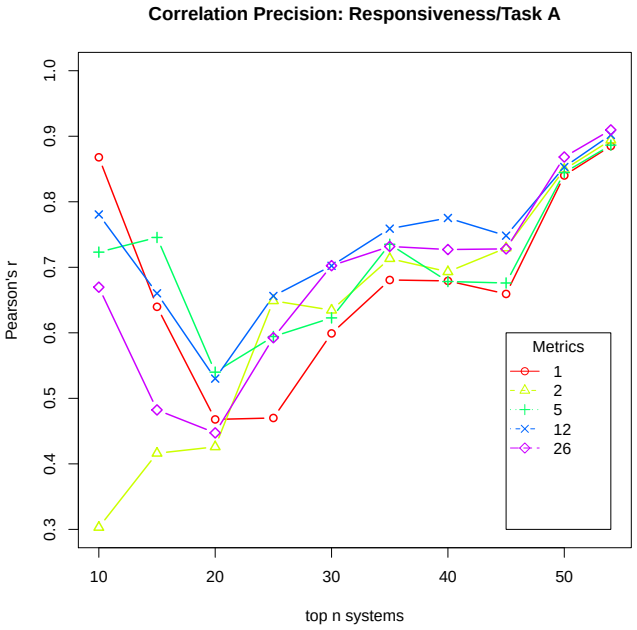

Figure 6. Correlations for n top systems sorted by evaluation metric for Task A

In future work, we want to explore bootstrapping approaches in order to received more statistically reliable result regarding the correlation coefficient for automatic metrics vs. manual metrics

References

[1] R. Barzilay and M. Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008.

[2] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring Semantic Similarity between Words Using Web Search Engines. In *Proc. of 16th International World Wide Web Conference (WWW 2007)*, pages 757–766, Banff, Canada, 2007.

[3] M. Burrows, M. Burrows, D. Wheeler, and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical report, Digital SRC Research Report, 1994.

[4] J. Conroy, J. Schlesinger, and D. O'Leary. CLASSY 2009: Summarization and Metrics. In *Proceedings of the Second Text Analysis Conference (TAC 2009)*. NIST, 2009.

[5] G. Giannakopoulos, V. Karkaletsis, G. Vouros, and P. Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *ACM Trans. Speech Lang. Process.*, 5(3):1–39, 2008.

[6] B. J. Grosz, S. Weinstein, and A. K. Joshi. Centering: a framework for modeling the local coherence of discourse. *Comput. Linguist.*, 21(2):203–225, 1995.

[7] E. Hovy, C.-Y. Lin, and L. Zhou. Evaluating DUC 2005 using Basic Elements. In *Proceedings of Document Understanding Conference (DUC 2005)*, Vancouver, B.C., Canada, 2005.

[8] T. Joachims. Optimizing search engines using clickthrough data. In *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142. ACM SIGKDD, ACM, 2002.

[9] H. Kaplan, S. Landau, and E. Verbin. A simpler analysis of burrows–wheeler-based compression. *Theor. Comput. Sci.*, 387(3):220–235, 2007.

[10] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi. The Similarity Metric. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 50(12):3250–3264, 2004.

[11] S. Li, Y. Ouyang, W. Wang, and B. Sun. Multi-document summarization using support vector regression. In *Proceedings of DUC 2007, Rochester, USA*, 2007.

[12] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10, 2004.

[13] C.-Y. Lin and E. Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[14] A. Louis and A. Nenkova. Automatic summary evaluation without human models. In *Proceedings of Text Understanding Conference (TAC 2008)*, 2008.

[15] F. Schilder and R. Kondadadi. FastSum: Fast and Accurate Query-based Multi-document Summarization. In *Pro-*

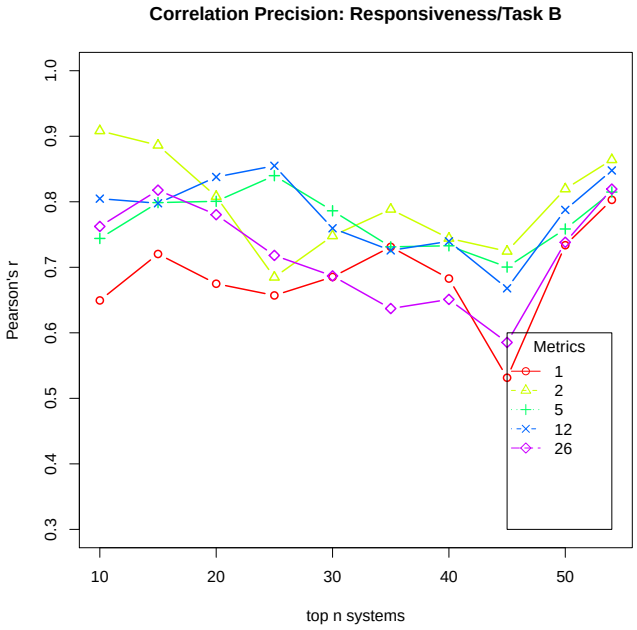

Figure 7. Correlations for n top systems sorted by evaluation metric for Task B

ceedings of ACL-08: HLT, Short Papers, pages 205–208, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[16] F. Schilder and R. Kondadadi. A metric for automatically evaluating coherent summaries via context chains. In *Proceedings of the 3rd IEEE International Conference on Semantic Computing (ICSC 2009)*, pages 65–70. IEEE Computer Society, 2009.

[17] F. Schilder, R. Kondadadi, J. L. Leidner, and J. G. Conrad. Thomson Reuters at TAC 2008: Aggressive Filtering with FastSum for Update and Opinion Summarization. In *Proceedings of the First Text Analysis Conference (TAC)*, Gaithersburg, MD, 2008. NIST.

[18] S. Tratz and E. Hovy. Summarization evaluation using transformed basic elements. Proceedings of Text Understanding Conference (TAC), 2008.