On alternative *automated* content evaluation measures

Rahul Katragadda
Team: *PRaSa*
Language Technologies Research Center
IIIT Hyderabad
rahul_k@research.iiit.ac.in

Abstract

In this draft we describe our TAC submissions and post-TAC experiments for Automated Evaluation of Summaries of Peers task of Text Analysis Conference (TAC). We approached the problem using two different approaches. Firstly, we use a generative modeling based approach to capture the sentence level presence of keywords in peer summaries and provide two fairly simple alternatives to identify keywords. Secondly, we used the Stanford dependency (SD) formalism to obtain a dependency recall based metric for summary evaluation. Our results show that the generative modeling approach is indeed promising and further investigation of keyword identification would obtain better results. For the Stanford-dependency based evaluation, performance has been similar to other dependency based evaluations of the likes of Basic Elements (BE) and DEPEval.

1 Introduction

Evaluation is crucial component in the area of automatic summarization; it is used both to rank multiple participant systems in a shared tasks, such as the summarization track at TAC 2009, 2008 and its DUC predecessors, and to developers whose goal is to improve the summarization systems. Summarization evaluation, as has been the case with other language understanding technologies, can foster the creation of reusable resources and infrastructure; it creates an environment for comparison and replication of results; and it introduces an element of competition to produce better results [3]. However, manual evaluation of a large number of documents necessary for a relatively unbiased view is often unfeasible, especially since multiple evaluations are needed in future to track incremental improvement in systems. Therefore, there is an urgent need for reliable automatic metrics that can perform evaluation in a fast and consistent manner.

Summarization Evaluation, like Machine Translation (MT) evaluation (or any other NLP systems' evaluation), can be broadly classified into two categories [4]. The first, an *intrinsic* evaluation, tests the summarization system in itself. The second, an *extrinsic* evaluation, tests the summarization system based on how it affects the completion of some other task. In the past *intrinsic*

evaluations have assessed mainly informativeness and coherence of the summaries. Meanwhile, *extrinsic* evaluations have been used to test the impact of summarization on tasks like reading comprehension, relevance assessment, etc.

2 Current Summarization Evaluations

In the Text Analysis Conference (TAC) series and the predecessor, the Document Understanding Conferences (DUC) series, the evaluation of summarization quality was conducted using both manual and automated metrics. Manual assessment, performed by human judges centers around two main aspects of summarization quality: *informativeness/content* and *readability/fluency*. Since manual evaluation is still the undisputed gold standard, both at TAC and DUC there was a phenomenal effort to evaluate manually as much data as possible.

Content Evaluations The content or informativeness of a summary has been evaluated based on various manual metrics. Earlier, NIST assessors used to rate each summary on a 5-point scale based on whether a summary is "very poor" to "very good". Since 2006, NIST uses the Pyramid framework to measure content responsiveness. In the pyramid method as explained in [9], assessors first extract all possible "information nuggets" or Summary Content Units *(SCUs)* from human-produced model summaries on a given topic. Each SCU has a weight associated with it based on the number of model summaries in which this information appears. The final score of a peer summary is based on the recall of nuggets in the peer.

All forms of manual assessment is time-consuming, expensive and not repeatable; whether scoring summaries on a Likert scale[1] or by evaluating peers against "nugget pyramids" as in the pyramid method. Such assessment doesn't help system developers — who would ideally like to have fast, reliable and most importantly *automated* evaluation metric that can be used to keep track of incremental improvements in their systems. So despite the strong manual evaluation criterion for informativeness, time tested automated methods viz. ROUGE, Basic Elements(BE) have been regularly employed to test their correlation with manual evaluation metrics like '*modified pyramid score*', '*content responsiveness*' and '*overall responsiveness*' of a summary. The creation and testing of automatic evaluation metrics is therefore an important research avenue. The goal is to create automated evaluation metrics that correlate very highly with these manual metrics.

3 Automated Content Evaluations

Based on the arguments set above, automated evaluation of content and form are necessary for tracking the developers incremental improvements, and a focused task on creation of automated metrics for content and form would help in the process. This was precisely the point being addressed at the TAC AESOP (Automatically Evaluating Summaries of Peers) task. In TAC 2009, AESOP task involves only "Automated Evaluation of Content and Responsiveness", and this paper addresses the same.

[1] http://en.wikipedia.org/wiki/Likert_Scale

In its first edition of AESOP task at TAC 2009, the purpose of the task was to promote research and development of systems that evaluate the quality of content in the summaries. The output of the automated metrics are compared against two manual metrics: *(modified) pyramid score*, which measures summary content and *overall responsiveness*, which measures a combination of content and linguistic quality.

Experimental Configuration In TAC 2009 task, for each topic there are 4 reference summaries and 55 peer summaries. The task output is to generate, for each peer summary, a score representing (in the semantics of the metric) the goodness of the summary content, measured against or without the use of model summaries.

4 Approach

In this work we report our submissions at TAC AESOP tasks and the following post-TAC experiments. We followed two major different approaches to our work. Firstly, we examined a generative modeling based approach to summarization evaluation where we modeled the amount of signature-terms being captured by peer summaries at sentence level based on how they are distributed in the source documents. Secondly, we used a dependency framework to measure dependency recall in comparison with reference summaries.

4.1 Generative Modeling of Reference Summaries

[5] describe two models based on the '*generative modeling framework*': a binomial model and a multinomial model, which they used to show that automated systems are being *query-biased* to be able to perform better on ROUGE like surface metrics. Our approach is to use the same generative models to evaluate summaries. We describe in the following sections, how various features extracted from reference summaries can be used in modeling how strongly peer summaries are able to imitate reference summaries.

We use generative modeling to model the distribution of *signature terms* in the source and the "likelihood of a summary being biased towards these *signature terms*". In the following sections we describe the two models of generative modeling, Binomial and Multinomial models.

4.1.1 Binomial Model

Let us consider there are 'k' words that we consider signature terms, as identified by any of the methods described in Section 4.2. The sentences in the input document collection are represented as a binomial distribution over the type of sentences. Let $C_i \in \{C_0, C_1\}$ denote classes of sentences without and with those '*signature terms*' respectively. For each sentence $s \in C_i$ in the input collection, we associate a probability $\mathrm{p}(C_i)$ for it to be emitted into a summary.

The likelihood of a summary then is :

$$L[summary; p\,(C_i)] = \frac{N!}{n_0!n_1!} p\,(C_0)^{n_0}\, p\,(C_1)^{n_1} \tag{1}$$

Where N is the number of sentences in the summary, and $n_0 + n_1 = \text{N}$; n_0 and n_1 are the cardinalities of C_0 and C_1 in the summary.

4.1.2 Multinomial Model

Previously, we described the binomial model where we classified each sentence into two classes, as being biased towards a *signature term* or not. However, if we were to quantify the amount of *signature-term bias* in a sentence, we associate each sentence to one among k possible classes leading to a multinomial distribution. Let $C_i \in \{C_0, C_1, C_2, \ldots, C_k\}$ denote the k levels of *signature-term bias*. C_i is the set of sentences, each having i signature terms.

The number of sentences participating in each class varies highly, with C_0 bagging a high percentage of sentences and the rest $\{C_1, C_2, \ldots, C_k\}$ distributing among themselves the rest sentences. Since the distribution is highly-skewed to the left, distinguishing systems based on log-likelihood scores using this model is easier and perhaps more accurate.

The likelihood of a summary then is :

$$L[summary; p(C_i)] = \frac{N!}{n_0! n_1! \cdots n_k!} p(C_0)^{n_0} p(C_1)^{n_1} \cdots p(C_k)^{n_k} \qquad (2)$$

Where N is the number of sentences in the 'peer summary', and $n_0 + n_1 + \cdots + n_k = \text{N}$; n_0, n_1,$\cdots$,n_k are respectively the cardinalities of C_0, C_1, $\cdots$,C_k, in the summary.

4.2 Signature Terms

The likelihood of certain characteristics ξ based on the binomial or multinomial model shows how well certain characteristics (ξ) of the input have been captured in a summary. For our approach, we need to have certain keywords from the reference summaries that are considered to be very important for the topic/query combination. We choose multiple alternative methods to identify such signature-terms. Here we list these methods:

1. Query terms
2. Model consistency
3. Part-Of-Speech (POS)

4.2.1 Query Terms

If we consider *query terms* as the characteristics that discriminate important sentences from unimportant ones, we obtain the likelihood of a summary emitting a *query-biased* sentence. Earlier, [5] have shown that such a likelihood has very high system-level correlation with ROUGE scores. Since ROUGE correlates very highly with manual evaluations ('*pyramid evaluation*' or '*overall responsiveness*'), a naïve assumption is that likelihood modeling of *query-bias* would correlate well with manual evaluations. This assumption led us to use this method as a baseline for our experiments. Our baselines for this work have been explained in Section 5.

4.2.2 Model Consistency

The hypothesis behind the method is that a term is important if it is part of a reference summary. In this method we obtain all the terms that are commonly agreed upon by reference summaries. The idea is that the more the reference summaries agree the more important they are. This is based on the assumption that word level importance sums up towards sentence inclusion. Since there are 4 reference summaries available for each topic, we can use reference agreement in two ways:

- total agreement
- partial agreement

Total agreement In the case of *total agreement*, only the words that occur in all reference summaries are considered to be important. This case leads to only a single run which we would call '*total-agreement*'.

Partial agreement In the case of *partial agreement*, words that occur in at least 'k' reference summaries are considered to be important. Since there are 4 reference summaries per topic, a term would be considered a '*signature term*' if it occurs in 'k' of those 4 reference summaries. There were a total of 3 runs in this case : '*partial-agreement-1*', '*partial-agreement-2*' and '*partial-agreement-3*'.

4.2.3 POS Features

We hypothesized that a certain type of words (or parts-of-speech) could be more informative than the other words, and that in modeling their occurrence in peer summaries we are defining informativeness of the peers with respect to models.

Part-of-speech tagger Traditional grammar classifies words based on eight *parts of speech*: the verb, the noun, the adjective, the pronoun, the adverb, the preposition, the conjunction and the interjection. Each part of speech explains not what the word is, but how the word is used. Infact the same word can be a noun in one sentence and a verb or adjective in another. We have used the Penn Treebank Tag-set [6] for our purposes. For automated tagging we have used the Stanford POS tagger [11, 10] in these experiments.

Tag Subset Selection – feature selection Based on an analysis of how each 'POS tag' performs at the task we selectively combine the set of features. We used the following 'POS tag' features: *NN*, *NNP*, *NNPS*, *VB*, *VBN*, *VBD*, *CD*, *SYMB*, and their combinations. We experimented with a lot of combinations of these features and zeroed on to a final list of combinations that form the runs described in this work. The final list of runs comprises of some of the individual 'POS tag' features and some combinations, they are:

- NN
- NNP
- NNPS
- NOUN – A combination of NN, NNP and NNPS features.
- VB
- VBN

- VBD
- VERB – A combination of VB, VBN and VBD features.
- CD
- SYMB
- MISC – A combination of CD and SYMB features.
- ALL – A combination of NOUN, VERB and MISC features.

4.3 Stanford Dependencies

A *dependency parse* represents dependencies between individual words. A *typed dependency parse* additionally labels dependencies with grammatical relations such as *subject* and *indirect object*. The *Stanford typed dependencies* (SD) are one such formalism which are based on grammatical relations loosly defined on [1]. The grammatical relations are arranged in a hierarchy, with *dependent* as the most generic relation at the root. The selection of grammatical relations to be included in SD was motivated by practical rather than theoretical concerns. The motivation behind the typed Stanford dependencies representation is given in [7, 2] while a detailed description of Stanford dependencies are given in [8].

We used the collapsed typed dependencies for our run using Stanford dependencies. Our run, *sd-recall* determines the number of dependencies recalled by a peer summary when compared against all the reference summaries. Our evaluation method is similar to BE and DEPEval in that it compares two unordered sets of dependencies.

$$SD\text{-}recall = \frac{|D_{cand}| \bigcap |D_{ref}|}{|D_{ref}|}$$

Where D_{cand} are the set of candidate dependencies and D_{ref} are the sent of reference dependencies. We haven't experimented with variations of SD-recall till the time of submission of this report and SD-recall reported later in results is the basic version described here. It must, however, be noted that we considered all the dependencies (even the trivial ones) and that we used the collapsed version of typed dependencies. So it is possible that we were comparing two reasonably matching dependencies but still having a mismatch. Also, we haven't considered partial matching it is difficult to expect that the matching has been accurate because when in doubt the Stanford dependency software falls back to a typed dependency above in the hierarchy.

5 Experiments and Evaluations

Our experimental setup was primarily defined based on how signature terms have been identified. We have detailed few methods of identification of signature-terms in Section 4.2. For each method of identifying *signature terms* we have 1 or more runs as described earlier.

Baselines. Apart from the set of runs described in Section 4.2, we propose to use the following two baselines.

- Binomial modeling for *query terms*.
- Multinomial modeling for *query terms*.

Finally we have one run based on the dependency framework, described earlier in the paper.

Datasets The experiments shown here were performed on TAC 2009 update summarization datasets which have 44 topics and 55 system summaries for each topic apart from 4 human reference summaries. And since in our methods there is no clear way to distinguish evaluation of cluster A's or cluster B's summary – we don't evaluate the update of a summary – we effectively have 88 topics to evaluate on.

Evaluations Evaluation of these *evaluation metrics* is done based on how well these new metrics correlate with manual evaluations. This task, despite the complexity involved, boils down to a simpler problem, that of information ordering. We have a reference ordering and have various metrics that provide their own ordering for these systems. Comparing an ordering of information with another is a fairly well understood task and we would use correlations between these manual metrics and the metrics we proposed in this work to show how well our metrics are able to imitate human evaluations in being able to generate similar ordering of systems. We use Pearson's Correlation Coefficient of system level average scores produced by all systems based on our metrics and by the manual methods.

RUN	**Pyramid**		**Responsiveness**	
	AllPeers	**NoModels**	**AllPeers**	**NoModels**
High Baselines				
ROUGE-SU4	0.734	0.921	0.617	0.767
Basic Elements (BE)	0.586	0.857	0.456	0.692
Baselines				
Binom(query)	0.217	0.528	0.163	0.509
Multinom(query)	0.117	0.523	0.626	0.514
Experimental Runs				
POS based				
NN	0.909	0.867	0.853	0.766
NNP	0.666	0.504	0.661	0.463
NOUN	0.923	0.882	0.870	0.779
VB	0.913	0.820	0.877	0.705
VBN	0.931	0.817	0.929	0.683
VBD	0.944	0.859	0.927	0.698
VERB	0.972	0.902	0.952	0.733
CD	0.762	0.601	0.757	0.561
MISC	0.762	0.601	0.757	0.561
ALL	0.969	0.913	0.934	0.802
Model Consistency/Agreement				
total-agreement	0.727	0.768	0.659	0.682
partial-agreement-3	0.867	0.856	0.813	0.757
partial-agreement-2	0.936	0.893	0.886	0.791
partial-agreement-1	0.966	0.895	0.930	0.768
Dependency Based Evaluations				
SD-recall	0.640	0.869	0.516	0.707

Table 1: Correlation scores for Cluster A

6 Results

Our target for these focused experiments were to create alternatives to the content evaluation metrics (*pyramid method* and *overall responsiveness*), that

RUN	Pyramid		Responsiveness	
	AllPeers	**NoModels**	**AllPeers**	**NoModels**
High Baselines				
ROUGE-SU4	0.586	0.940	0.564	0.729
Basic Elements (BE)	0.629	0.924	0.447	0.694
Baselines				
Binom(query)	0.210	0.364	0.178	0.372
Multinom(query)	-0.004	0.361	-0.020	0.446
Experimental Runs				
POS based				
NN	0.908	0.845	0.877	0.788
NNP	0.646	0.453	0.631	0.380
NOUN	0.909	0.848	0.878	0.783
VB	0.872	0.871	0.875	0.742
VBN	0.934	0.873	0.944	0.720
VBD	0.922	0.909	0.914	0.718
VERB	0.949	0.951	0.942	0.784
CD	0.807	0.599	0.800	0.497
MISC	0.807	0.599	0.800	0.497
ALL	0.957	0.921	0.931	0.793
Model Consistency/Agreement				
total-agreement	0.811	0.738	0.808	0.762
partial-agreement-3	0.901	0.839	0.882	0.806
partial-agreement-2	0.949	0.898	0.924	0.817
partial-agreement-1	0.960	0.903	0.936	0.763
Dependency Based Evaluations				
SD-recall	0.647	0.918	0.465	0.685

Table 2: Correlation scores for Cluster B

are either 'too expensive' or 'non-replicable' or both. It is unlikely that any single automated evaluation measure would be able to correctly reflect both readability and content responsiveness, since they represent form and content which are separate qualities of a summary and would need different measures. We chose to imitate content since having better content in a summary is more important than having a readable summary.

In Tables 1 and 2 we present system level Pearson's correlations between the scores provided by our metrics — as well as the time tested automated evaluation metrics ROUGE-SU4 and Basic Elements (BE) — and the manual Pyramid scores. The table also includes correlations with the manual *Overall Responsiveness* measure, which reflects both content and form; later we would observe that the correlations are much higher with respect to pyramids than with overall responsiveness, this is because in our approach we are trying to capture how well content of model summaries are being reciprocated in system summaries.

6.1 Discussion

We have used two separate settings for displaying results: an *AllPeers* case and a *NoModels* case. AllPeers case consists of the scores returned by the metric for all the summarizers (automated and human), while in the case of NoModels case only automated summarizers are scored using the evaluation metrics. This setup helps distinguish methods that are able to differentiate two things:

- Metrics that are able to differentiate humans from automated summarizers.
- Metrics that are able to rank human summarizers in the desired order.

Results[2] have shown that no single metric is good at distinguishing everything, however they also show that certain type of keywords have been instrumental in providing the key distinguishing power to the metric. For example, *VERB and NOUN* features have been key the contributors to *ALL* run. Also as an interesting side note we observe that having high number of 'significant' signature-terms seems to be better than a low number of *'strong' signature-terms*, as seen from the experiments on *total-agreement* and *partial-agreement*. The most important result of our approach has been that our method was very highly correlated with "overall responsiveness", which again is a very good sign for an evaluation metric.

7 Conclusion

In the context of TAC AESOP (Automatically Evaluating Summares Of Peers) task, we model the problem as an information ordering problem; our approach (and indeed others) should now be able to rank systems (and possibly human summarizers) in the same order as human evaluation would have produced. We show how a well known generative model could be used to create automated evaluation systems comparable to the state-of-the-art. Our method is based on a multinomial model distribution of key-terms (or signature terms) in document collections, and how they are captured in peers.

We have used two types of signature-terms to model the evaluation metrics. The first is based on POS tags of important terms in a model summary and the second is based on how much information the reference summaries shared among themselves. Our results show that verbs and nouns are key contributors to our best run which was dependent on various individual features. Another important observation was that all the metrics were consistent in that they produced similar results for both cluster A and cluster B (update summaries). The most startling result is that in comparison with the automated evaluation metrics currently in use (ROUGE, Basic Elements) our approach has been very good at capturing "overall responsiveness" apart from pyramid based manual scores.

References

[1] J. Carroll, G. Minnen, D. Pearce, Y. Canning, S. Devlin, and J. Tait. Simplifying text for language-impaired readers. 1999.

[2] Marie-Catherine de Marneffe and Christopher D. Manning. The stanford typed dependencies representation. In *CrossParser '08: Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

[2] We have excluded *NNPS* and *SYMB* from the analysis since they didn't have enough samples in the testset, so as to obtain consistent results.

[3] Lynette Hirschman and Inderjeet Mani. Evaluation. 2001.

[4] Karen Spärck Jones and Julia R. Galliers. *Evaluating Natural Language Processing Systems: An Analysis and Review*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.

[5] Rahul Katragadda and Vasudeva Varma. Query-focused summaries or query-biased summaries ? In *Proceedings of the joint conference of the 47th Annual meeting of the Association of Computational Linguistics and the 4th meeting of International Joint Conference on Natural Language Processing, ACL-IJCNLP 2009*. Association of Computational Linguistics, 2009.

[6] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, 19(2):313–330, 1993.

[7] M. Marneffe, B. Maccartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-06*, pages 449–454, 2006.

[8] M. Marneffe, B. Maccartney, and C. Manning. *Stanford typed dependencies manual*, 2008.

[9] Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. The pyramid method: Incorporating human content selection variation in summarization evaluation. In *ACM Trans. Speech Lang. Process.*, volume 4, New York, NY, USA, 2007. ACM.

[10] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[11] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, pages 63–70, Morristown, NJ, USA, 2000. Association for Computational Linguistics.