SSAS: Semantic Similarity for Abstractive Summarization

Raghuram Vadapalli **Litton J Kurisinkel** **Manish Gupta*** **Vasudeva Varma**
International Institute of Information Technology – Hyderabad
Hyderabad India
{raghuram.vadapalli, litton.jkurisinkel}@research.iiit.ac.in
{manish.gupta, vv}@iiit.ac.in

Abstract

Ideally a metric evaluating an abstract system summary should represent the extent to which the system-generated summary approximates the semantic inference conceived by the reader using a human-written reference summary. Most of the previous approaches relied upon word or syntactic sub-sequence overlap to evaluate system-generated summaries. Such metrics cannot evaluate the summary at semantic inference level. Through this work we introduce the metric of Semantic Similarity for Abstractive Summarization **(SSAS)**[1], which leverages natural language inference and paraphrasing techniques to frame a novel approach to evaluate system summaries at semantic inference level. SSAS is based upon a weighted composition of quantities representing the level of agreement, contradiction, topical neutrality, paraphrasing, and optionally ROUGE score between a system-generated and a human-written summary.

1 Introduction

Abstractive summarization techniques try to mimic human expert's capabilities of inference making and producing a summary in her own writing style. Automated abstractive summarization techniques are highly desirable since one needs a lot of effort and language skills for generating summaries from varying information sources such as social media, databases, web articles etc. It is crucial for a constructive evolution of research on abstractive summarization, to establish a metric which can judge the quality of a system-generated abstractive summary. An ideal metric should be able to represent the similarity of *semantic inference* perceived by a reader from system-generated summary to that from a human-written reference summary.

Most of the existing summarization metrics are well-suited for extractive summaries, and are directly or indirectly based upon word or syntactic substructure overlap (Lin, 2004). Evaluation of abstractive summarization needs a semantic overlap based method. Although there are some metrics which attempt to evaluate the summary at semantic level (Nenkova and Passonneau, 2004; Passonneau et al., 2013; Yang et al., 2016), they either demand high level of human involvement or rely on external discrete vocabulary information (Miller et al., 1990). Also they are less equipped to conceive the accurate semantic inference from long sequences of summary text.

For instance, consider the following statements.
A: Mary lived through an era of liberating reform for women.
B: Mary's life spanned years of incredible change for women.
C: Mary lived through an era of suppression of women.

Considering *A* as the reference summary element, most of the previous metrics give higher score to *C* than *B* even when *C* is clearly contradicting *A*. Actual scores for above samples are shown in Table 1.

Bowman et al. (2015) mention that understanding entailment and contradiction is fundamental to understanding natural language. The lack of consideration of semantics when evaluating summarization automatically, motivates us to propose a new metric focused on semantic matching between system and human summaries.

* The author is also a Principal Applied Scientist at Microsoft (gmanish@microsoft.com).

[1] Data and code are shared at http://somewhereonweb.com.

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 198–203,
Taipei, Taiwan, November 27 – December 1, 2017

Method	B	C
ROUGE-1	0.33	0.66
ROUGE-2	0.0	0.4
ROUGE-L	0.33	0.66
ROUGE-SU4	0.05	0.45
PEAK	0.45	0.33
SSAS	0.65	0.48

Table 1: Scores given to Samples *B* and *C* by Various Metrics

Our main contributions are as follows.

- We propose a novel metric SSAS for semantic assessment of abstractive summaries.
- The method includes computing various semantic and lexical similarity measures between reference summary and system summary, and learning a weight vector to combine these measures into a single score such that the score maximally correlates with human evaluation of summaries.
- We experimentally show the robustness and effectiveness of SSAS.

The rest of the paper is organized as follows. Section 2 describes previous attempts at evaluating summarization systems. Section 3 describes our approach in detail. We discuss our experimental results in Section 4. We conclude with a summary in Section 5.

2 Related Work

The following are two broad approaches popular for evaluation of summaries proposed in the past.

2.1 ROUGE Based Approaches

ROUGE-n measures evaluate the system summary on the basis of n-gram overlap. ROUGE-SU and ROUGE-L evaluate content overlap in a more sophisticated manner. But they still cannot reliably capture semantic overlap or semantic contradiction. Recently Ng and Abrecht (2015) tried to enhance ROUGE using word embeddings, but word embeddings cannot grasp the semantic inference induced by a sequence of words.

2.2 Pyramid Based Approaches

In pyramid evaluation (Nenkova and Passonneau, 2004), Summarization Content Units (SCUs) are extracted from model summaries and they are given weights which are equal to the number of reference summaries they occur in. After this, a generated summary is given a score which is equal to the normalized sum of the weights of the overlapping SCUs. Pyramid score does not evaluate the semantic overlap in a continuous space, and also requires manual efforts when performing evaluation.

Autopyramid (Passonneau et al., 2013) automates a part of the pyramid based evaluation which checks whether an SCU is present in the generated summary. Though they use various generic dense representations (Guo and Diab, 2012) for estimating semantic similarity between SCUs, Autopyramid cannot explicitly quantify the quality of a summary based on its agreement or contradiction with a reference summary.

The PEAK (Yang et al., 2016) method for evaluation automates the extraction part of the SCUs, and they use the ADW (Align, Disambiguate and Walk) algorithm (Pilehvar et al., 2013) to compute semantic similarity. However, their approach fails to model contradiction, paraphrase identification and other features like natural language inference.

3 Approach for SSAS Computation

We first extract SCUs using the automatic SCU extraction scheme introduced by PEAK model I, which in turn relies on Open Information Extraction (OpenIE) (Angeli et al., 2015) for the extraction process. Given a reference summary R and a system summary S, we obtain SCU sets $SCUs(R)$ and $SCUs(S)$ with cardinality n and m respectively. Next, we derive a set of natural language inference and paraphrasing features from the text pieces. Computation of these features is explained in Section 3.1. After that, we use a ranking model to learn the weights for combining these features to obtain a score. Finally, we normalize the obtained score. Ranking and Normalization are discussed in detail in Section 3.2.

3.1 Features for SSAS

SSAS uses natural language inference (NLI) features, paraphrase features and ROUGE-SU4 as features. We discuss these in detail below.

3.1.1 NLI Features

In this subsection, we consider features that capture natural language inference-based similarity between text pieces. We leverage the neural attention model, proposed by Cheng et al. (2016)

for this purpose. Let $E(a,b), C(a,b)$ and $N(a,b)$ be the entailment, contradiction and topic neutrality probabilities respectively between two SCUs a and b such that $E(a,b) + C(a,b) + N(a,b) = 1$. Based on these probability scores, we compute values for the following features from sets $SCUs(R)$ and $SCUs(S)$.

Combined Entailment Scores: We compute two features F_{e1} and F_{e2} which quantify the combined entailment score between reference summary and system summary. F_{e1} (Eq. 1) finds the SCU $b \in SCUs(S)$ that is best entailed by each SCU $a \in SCUs(R)$, and then computes average entailment score across all $a \in SCUs(R)$. F_{e2} (Eq. 2) is defined similarly, but aggregates scores across all $a \in SCUs(S)$ and considers entailment of an SCU $b \in SCUs(R)$ by an SCU $a \in SCUs(S)$.

$$F_{e1} = \frac{1}{n} \sum_{a \,\epsilon\, SCUs(R)} \max_{b \epsilon SCUs(S)} E(a,b) \quad (1)$$

$$F_{e2} = \frac{1}{m} \sum_{a \,\epsilon\, SCUs(S)} \max_{b \epsilon SCUs(R)} E(a,b) \quad (2)$$

Combined Contradiction Scores: Similar to entailment scores, two features F_{c1} and F_{c2} quantify the combined contradiction score as shown in Eqs. 3 and 4 respectively.

$$F_{c1} = \frac{1}{n} \sum_{a \,\epsilon\, SCUs(R)} \max_{b \epsilon SCUs(S)} C(a,b) \quad (3)$$

$$F_{c2} = \frac{1}{m} \sum_{a \,\epsilon\, SCUs(S)} \max_{b \epsilon SCUs(R)} C(a,b) \quad (4)$$

Combined Topic Neutrality Scores: Finally, two features F_{n1} and F_{n2} are computed to quantify the combined topical neutrality score as shown in Eqs. 5 and 6 respectively.

$$F_{n1} = \frac{1}{n} \sum_{a \,\epsilon\, SCUs(R)} \max_{b \epsilon SCUs(S)} N(a,b) \quad (5)$$

$$F_{n2} = \frac{1}{m} \sum_{a \,\epsilon\, SCUs(S)} \max_{b \epsilon SCUs(R)} N(a,b) \quad (6)$$

3.1.2 Paraphrase Features

We compute the paraphrasing probability $P(a,b)$ for two SCUs a and b using the model proposed by Kiros et al. (2015) which is trained on MSRP corpus (Bouamor et al., 2012). The combined paraphrase scores F_{p1} and F_{p2} are given by Eqs. 7 and 8 respectively.

$$F_{p1} = \frac{1}{n} \sum_{a \,\epsilon\, SCUs(R)} \max_{b \epsilon SCUs(S)} P(a,b) \quad (7)$$

$$F_{p2} = \frac{1}{m} \sum_{a \,\epsilon\, SCUs(S)} \max_{b \epsilon SCUs(R)} P(a,b) \quad (8)$$

3.1.3 ROUGE-SU4 Feature

Along with other dense semantic level features, n-gram overlap can also be indicative to evaluate the summary quality. Following this intuition, we include ROUGE score between the system summary S and the reference summary R as one of the features.

$$F_R = \text{ROUGE-SU4}(S,R) \quad (9)$$

3.2 Computing SSAS

For every pair (R,S), we concatenate the extracted features to form the feature vector $\vec{\boldsymbol{f}}$.

$$\vec{\boldsymbol{f}} = [F_{e1}, F_{e2}, F_{c1}, F_{c2}, F_{n1}, F_{n2}, F_{p1}, F_{p2}, F_R]$$

We estimate a score for S with respect to R as shown in Eq. 10.

$$score(S,R) = \vec{\boldsymbol{\lambda}} \cdot [\vec{\boldsymbol{f}}, \mathbf{1}] \quad (10)$$

where $\vec{\boldsymbol{\lambda}}$ is a learned 10 dimensional parameter vector (9 for features + 1 for bias). The value of $\vec{\boldsymbol{\lambda}}$ is optimized so that the score as computed using Eq. 10 matches human assigned score. Finally, we compute the normalized SSAS score for a system summary with respect to the reference summary using min-max normalization as shown in Eq. 11.

$$SSAS(S,R) = \frac{\vec{\boldsymbol{\lambda}} \cdot ([\vec{\boldsymbol{f}}, \mathbf{1}] - [\vec{\boldsymbol{f_{min}}}, \mathbf{1}])}{\vec{\boldsymbol{\lambda}} \cdot ([\vec{\boldsymbol{f_{max}}}, \mathbf{1}] - [\vec{\boldsymbol{f_{min}}}, \mathbf{1}])} \quad (11)$$

where $\vec{\boldsymbol{f_{max}}}$ is the feature vector of an ideal summary obtained by setting values for entailment, paraphrase and ROUGE features to 1, and rest all to 0. $\vec{\boldsymbol{f_{min}}}$ is the feature vector of an extremely bad summary obtained by setting values for contradiction features to 1, and rest all to 0. Overall, SSAS scores lie between 0 and 1. The higher the SSAS score, the better is the system summary S.

4 Experiments and Results

In this section, we discuss details of our dataset, comparison of multiple ranking models, and comparison of SSAS with other metrics for summarization.

4.1 Dataset

We obtained DUC[2] 2002, 2003, 2004 datasets and the TAC[3] dataset which contain triplets (D, HS, LS) where D denotes the document/corpus to be summarized, HS denotes the human-written summary of D, and LS denotes the list of system summaries with their corresponding human evaluation scores.

In total, we collected approximately 250 (D, HS, LS) triplets from these datasets. Since SSAS is framed to evaluate abstractive summary, we constructed a test set comprising strictly abstractive summaries for 30 documents in DUC 2002 dataset separate from the training and development sets. Although we perform experiments by taking single human-written summary, extension to use multiple reference summaries is straightforward. Using multiple references also aligns with the notion that there is no single best summary.

4.2 Learning to Rank by Optimizing $\vec{\lambda}$

The value of $\vec{\lambda}$ is optimized by applying 'Learning To Rank' (LTR) techniques over training data comprising of (D, HS, LS) triplets. The optimization is performed such that the ranking order obtained by Eq. 10 maximally correlates with ranking order obtained by human scores. We executed development experiments with the three different LTR techniques (Burges et al., 2005; Cao et al., 2007): Pairwise, Listwise and Regression.

The results on the development dataset in terms of Pearson correlation and Spearman rank correlation are shown in Table 2. As the table shows, Listwise comparison gained better results though the results are not significantly different from others.

4.3 Evaluating SSAS

As mentioned earlier, to create the test dataset we chose 30 document-reference summary pairs from DUC 2002 and asked two human annotators to read each of the reference summary and to reproduce the content in their own writing style with full freedom to choose the convenient vocabulary. The human summarizers are post-graduate students in computational linguistics and we call the summary written by them as abstract summary $AbSum$. For each document in the test set a random subset of sentences are chosen to form a random summary $RandSum$ of the document. We use SSAS to score $AbSum$ and $RandSum$ with respect to the reference summary. The reliability of the metric is proportional to the number of times $AbSum$ is scored better than $RandSum$.

For the metric to be reliable, it is important to show that it produces consistent results even if the training and test data are from different types of data. For this purpose, we trained the model on the following three different subsets and recorded

[2]http://duc.nist.gov/data.html

[3]http://tac.nist.gov/data/past/2011/Summ11.html

Method	Pearson Correlation	Spearman Rho
Pairwise	0.970	0.975
Listwise	0.978	0.979
Regression	0.964	0.967

Table 2: Pearson Correlation and Spearman Rho for the Three Ranking Models

Method	Accuracy	σ
ROUGE-1	0.810	NA
ROUGE-2	0.782	NA
ROUGE-L	0.825	NA
ROUGE-SU4	0.839	NA
PEAK	0.861	NA
SSAS without F_{e1}, F_{e2}	0.854	5.44e-6
SSAS without F_{c1}, F_{c2}	0.893	4.38e-6
SSAS without F_{n1}, F_{n2}	0.862	6.12e-6
SSAS without F_{p1}, F_{p2}	0.845	4.1e-6
SSAS without F_R	0.882	5.64e-6
SSAS with all features	**0.913**	6.22e-6
Human	1.000	NA

Table 3: Results on the Custom Dataset

Method	Execution Time (sec)
ROUGE	4
PEAK	10
SSAS	200

Table 4: Approximate Execution Times of Various Metrics on Test Data

standard deviation of accuracy: a subset of 80 triples from DUC, a subset of 80 triples from TAC, a subset of 40 triples each from DUC and TAC.

Table 3 shows the mean results obtained by repeatedly selecting sentences for random summary 10 times over each of the above training subsets and the standard deviation obtained on changing training subset. The insignificance of standard deviation shows that the metric produces consistent results as long as training data is reliable.

4.4 Analysis of Results

From Table 3, we see that excluding contradiction does not have much impact on results. This is not surprising as we trained the data on DUC and TAC datasets which have very few contradictory sentences as they are extractive summaries. Nonetheless, we propose incorporating contradiction as abstractive summaries have a good chance of producing contradictory sentences. We can address this problem by training on human-evaluated abstractive summaries which we leave as future work.

We also see that F_{p1} and F_{p2} (features corresponding to paraphrasing) are important features since excluding them has a significant impact on the results.

Table 4 shows the approximate execution times for various methods. Since SSAS performs similarity assessment using deep semantic analysis, it does take significantly large amount of execution time compared to other methods. However, SSAS computations for multiple summaries can be easily parallelized.

5 Conclusions

In this work, we proposed a novel metric SSAS for semantic assessment of abstractive summaries. Our experiments show that SSAS outperforms previously proposed metrics. While the metric shows a very strong correlation with human judgments, it is computationally very intensive because of the deep semantic models which are used to compute various features. In the future, we plan to explore more efficient ways to obtain the feature vectors for SSAS computation.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging Linguistic Structure For Open Domain Information Extraction. In *Proc. of the* 53^{rd} *Annual Meeting of the Association for Computational Linguistics (ACL)*. pages 344–354.

Houda Bouamor, Aurélien Max, and Anne Vilnat. 2012. Automatic Acquisition and Validation of Subsentential Paraphrases: A Bilingual Study. *Journal of the Association pour le Traitement Automatique des Langues (TAL)* 53(1):11–37.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A Large Annotated Corpus for Learning Natural Language Inference. In *Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to Rank using Gradient Descent. In *Proc. of the* 22^{nd} *Intl. Conf. on Machine learning (ICML)*. pages 89–96.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proc. of the* 24^{th} *Intl. Conf. on Machine learning (ICML)*. pages 129–136.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long Short-Term Memory Networks for Machine Reading. *Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing (EMNLP)* pages 551–562.

Weiwei Guo and Mona T. Diab. 2012. Modeling Sentences in the Latent Space. In *Proc. of the* 50^{th} *Annual Meeting of the Association for Computational Linguistics (ACL)*. pages 864–872.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Proc. of the 2015 Conf. on Advances in Neural Information Processing Systems (NIPS)*. pages 3294–3302.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proc. of the Workshop on Text Summarization Branches Out (WAS)*. pages 74–81.

G. A. Miller, C. Fellbaum, D. Gross, and K. J. Miller. 1990. Introduction to WordNet: An On-Line Lexical Database. *Intl. Journal of Lexicography (IJL)* 3(4):235–244.

Ani Nenkova and Rebecca J. Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method. In *Proc. of the 2004 Human Language Technology Conf. of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*. pages 145–152.

Jun-Ping Ng and Viktoria Abrecht. 2015. Better Summarization Evaluation with Word Embeddings for

ROUGE. In *Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1925–1930.

Rebecca J Passonneau, Emily Chen, Weiwei Guo, and Dolores Perin. 2013. Automated Pyramid Scoring of Summaries using Distributional Semantics. In *Proc. of the* 51^{st} *Annual Meeting of the Association for Computational Linguistics (ACL)*. pages 143–147.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity. In *Proc. of the* 51^{st} *Annual Meeting of the Association for Computational Linguistics (ACL)*. pages 1341–1351.

Qian Yang, Rebecca J Passonneau, and Gerard de Melo. 2016. PEAK: Pyramid Evaluation via Automated Knowledge Extraction. In *Proc. of the* 13^{th} *AAAI Conf. on Artificial Intelligence (AAAI)*. pages 2673–2680.