Pyramid-based Summary Evaluation Using Abstract Meaning Representation

Josef Steinberger[1,2] , **Peter Krejzl**[2] , and **Tomáš Brychcín**[1,2]

[1]NTIS – New Technologies for the Information Society,
Faculty of Applied Sciences, University of West Bohemia, Czech Republic
[2]Department of Computer Science and Engineering,
Faculty of Applied Sciences, University of West Bohemia, Czech Republic
{jstein,krejzl,brychcin}@kiv.zcu.cz

Abstract

We propose a novel metric for evaluating summary content coverage. The evaluation framework follows the Pyramid approach to measure how many summarization content units, considered important by human annotators, are contained in an automatic summary. Our approach automatizes the evaluation process, which does not need any manual intervention on the evaluated summary side. Our approach compares abstract meaning representations of each content unit mention and each summary sentence. We found that the proposed metric complements well the widely-used ROUGE metrics.

1 Introduction

Evaluating the quality of automatic summaries is a known problem, which has attracted many computational linguists. Summary quality has different aspects, mainly the readability and the content coverage. Readability scores the summary by its grammaticality, non-redundancy, referential clarity, focus, structure or coherence. For many use cases (e.g. skimming through a summary of related news articles), content coverage decides whether the summary is useful or not.

The Pyramid method (Nenkova et al., 2007) is a well-designed framework for measuring content coverage. It computes how many summarization content units (SCUs) are shared between model summaries and the target system summary. The matching cannot be done on the lexical level, because the same information is often expressed in different words. In the past DUC/TAC[1] challenges the matching was done by humans. Although the participants received a precise feedback on their summaries, it was not possible to evaluate a new set of summaries after the evaluation exercise.

There has been many approaches to perform a fully automatic evaluation, including the most widely used ROUGE (Lin, 2004). Although they can rank the systems well enough on the whole corpus, their correlation with human judgements on the summary level is limited.

Abstract meaning representation (AMR) was introduced by Banarescu et al. (2013). AMR is intended to abstract meaning away from syntax. Sentences, which are similar in meaning should be assigned the same AMR, even if they are not identically written. The abstractive ability makes it very suitable for summarization, already shown by Liu et al. (2015). And as the pyramid matching requires dealing with paraphrases, it led us to the idea to use AMR for automatic pyramid evaluation.

Next, we discuss the past summarization shared tasks and evaluation metrics. An overview of the AMR format, a state-of-the-art parser, and an AMR graph similarity metric are described in Section 3. In Section 4, we define a novel metric, denoted as APE[2]. We evaluate the metric and compare/combine it with other metrics in Section 5.

2 Related Work

The most widely used automatic content evaluation metric is ROUGE (Lin and Hovy, 2003; Lin, 2004). It measures the word n-gram overlap between evaluated summary (*peer*) and the reference summaries (*models*, ideally written by humans). DUC/TAC evaluations showed that ROUGE correlates well with human judgements when scores of each peer are averaged over all topics of a corpus (Pearson's r is greater than .9).On the other hand, when as-

[1]Document Understanding Conference (DUC), followed by Text Analysis Conference (TAC), was a series of summarization shared tasks initiated by NIST (Over et al., 2007).

[2]APE denotes using **A**MR for **P**yramid-based **E**valuation.

Proceedings of Recent Advances in Natural Language Processing, pages 701–706,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_090

sessing individual summaries, correlation drops to ~ .6).ROUGE-2 (bigrams) and ROUGE-SU4 (bigrams with skip distance up to 4 words) correlated best on most of the corpora. Basic Elements (Hovy et al., 2005; Tratz and Hovy, 2008) were designed to address the problem of variable-size of semantic units by scoring syntactically coherent units. Head-modifier (BE-HM) pairs performed the best.

In the first DUC conferences, only a manual scale-based evaluation method was used. Later, it was complemented by the Pyramid method. The pyramid approach involves two tasks. First, human annotators identify SCUs, sets of text fragments that express the same basic content, in model summaries and create a pyramid (SCUs are weighted according to the number of models, in which they appear). Second, they evaluate a new summary against the pyramid. The *pyramid score* is computed by the total weight of all SCUs present in the candidate divided by the total SCU weight possible for an average-length summary (Nenkova et al., 2007). Although, creating a pyramid automatically would be useful for creating a new evaluation corpora, having an automated scoring of a new summary against a human-produced pyramid would make the method far more useful for developers. They would use a standard corpus and evaluate different versions of summarizers.

Several approaches for automating the pyramid scoring has been proposed. Harnly et al. (2005) tested several similarity metrics for matching SCUs against summaries and achieved the best results with unigram overlaps and single-linkage clustering. Passonneau et al. (2013) added two semantic similarities: a string comparison and a distributional semantics method, which performed better.

Emulating the pyramid method was one of the goals of the past TAC AESOP tasks in 2009 (Dang and Owczarzak, 2009), 2010 (Owczarzak and Dang, 2010), and 2011 (Owczarzak and Dang, 2011; Owczarzak et al., 2012). In 2009, several submissions achieved high correlations on the summarizer level. In 2010, ROUGE metrics were ranked very high, again based on summarizer-level correlations. In 2011, summary-level correlations revealed a room for improvement as the best Pearson correlation was .752 (Giannakopoulos and Karkaletsis, 2011), followed by ROUGE-SU4 (.736). Many models have been already proposed for the pyramid matching (Harnly et al., 2005; Passonneau et al., 2013), but the use of semantic relations between the sentence units has not been discussed yet.

3 Abstract Meaning Representation

Graph-structured semantic representations enable a direct semantic analysis of sentences. Banarescu et al. (2013) started annotating the logical meaning of sentences into AMR. In a nutshell, AMR graph is a rooted, labeled, directed acyclic graph, comprising a sentence. It is intended to abstract away from syntax and incorporates semantic roles, coreference, questions, modality, negation, and further linguistic phenomena[3].

Formally, we define a set of all possible triples:

$$\boldsymbol{T} = \left\{ (r, n_1, n_2) \;\middle|\; \begin{array}{l} r = \text{relation type} \\ n_1 = \text{variable} \\ n_2 = \text{variable or concept} \end{array} \right\} \quad (1)$$

AMR is then a subset of $\boldsymbol{T}$: $\boldsymbol{A} \subseteq \boldsymbol{T}$.

An example sentence, logical triples and corresponding AMR graph is illustrated in Figure 1 (left part). AMR introduces variables (graph nodes) for entities, events, properties, and states. Each node in the graph represents a semantic concept. These concepts can either be English words (*boy*), PropBank framesets (*want-01*) (Palmer et al., 2005), or special keywords. The *instance* relation assigns to each concept a variable which can be reused in other triples. Edge labels denote the relations that hold between entities (e.g. *ARG-0* relation between *want-01* and *boy*). The AMR triples take one of these forms: *relation*(*variable*, *concept*) or *relation*(*variable*$_1$, *variable*$_2$).

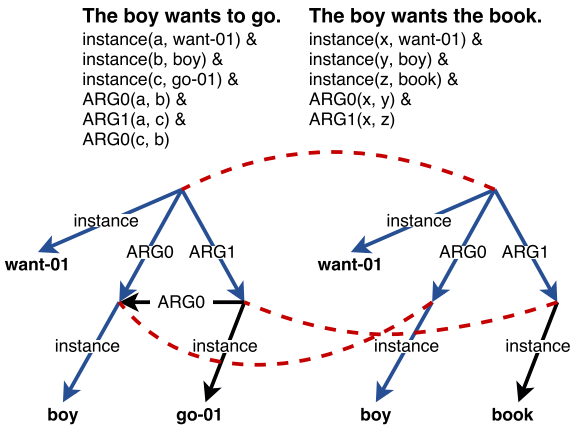

Figure 1: AMR trees and semantic matching.

Flanigan et al. (2014) introduced the first approach to parse sentences into AMR, which re-

[3]The AMR language is described in more detail in the AMR annotation guidelines: http://amr.isi.edu/language.html.

quires algorithms for alignment, structured prediction, and statistical learning (JAMR[4]). In order to automatically parse English into AMR, JAMR employs a two-part algorithm. First, it identifies the key concepts using a semi-Markov model. Second, it identifies the relations between the concepts by searching for the maximum spanning, connected subgraph, a similar approach to maximum spanning tree described in McDonald et al. (2005).

Cai and Knight (2013) introduced semantic match (Smatch), a metric that calculates the degree of overlap between two semantic feature structures. It first computes the maximum number of matching triples among all possible variable mappings. To obtain the best variable mapping, Smatch executes a brief search which uses integer linear programming and a hill climbing method (described by Cai and Knight (2013)). We define a function (*mmt*) which takes two AMRs, finds the optimal variable mapping, and returns the maximum number of matching triples:

$$\text{mmt} : \boldsymbol{T} \times \boldsymbol{T} \rightarrow \mathbb{Z}^{+} \tag{2}$$

Given the maximum number of matching triples, we can compute precision, recall, and F1 score, which is the Smatch score.

For our example (Figure 1), the highest scoring variable alignment is if $a = x$, $b = y$ and $c = z$. The blue-arrow relations are matched (4), the black-arrow relations are not matched (2 left, 1 right). Thus recall is $4/6$, precision is $4/5$. and the resulting Smatch score is .73.

4 APE – AMR-based Pyramid Evaluation

The aim is to measure how many SCUs identified by humans in a set of model summaries are contained in the target (evaluated) peer summary. Assume we have a sequence of C SCUs $\boldsymbol{c} = \{\boldsymbol{c}_i\}_{i=1}^{C}$. Their weights ($\lambda_i$) are based on the number of models, in which they were mentioned. Each SCU is described by a sequence of its mentions $\boldsymbol{c}_i = \{d_j\}_{j=1}^{D_i}$, where D_i is the number of models, in which SCU $\boldsymbol{c}_i$ is mentioned. Model summaries contain a sequence of M sentences $\boldsymbol{m} = \{m_i\}_{i=1}^{M}$, and the peer summary is a sequence of P sentences $\boldsymbol{p} = \{p_i\}_{i=1}^{P}$.

Every sentence from all models and from the peer is parsed by an AMR parser, resulting in a sequence of AMR trees. $\boldsymbol{A}_{m_i} \subseteq \boldsymbol{T}$ is the set of AMR triples of sentence m_i, similarly $\boldsymbol{A}_{p_i} \subseteq \boldsymbol{T}$ is the set of AMR triples of sentence p_i. As each SCU mention d_j is a part of a model sentence m_i, $\boldsymbol{A}_{d_j}$ contains only those triples of $\boldsymbol{A}_{m_i}$ whose constituents are contained in the SCU mention: $\boldsymbol{A}_{d_j} \subseteq \boldsymbol{A}_{m_i}$. These represent the "gold triples", which are searched in the peer summary sentences.

For each sentence in the peer summary, we get the corresponding AMR tree and match its triples against the gold ones. The Smatch variable mapping is used for getting a common set of triples (function mmt, see Equation 2). If there is a sentence that contains a larger percentage of gold triples than a threshold (τ), the SCU is considered covered by the peer summary. Note that compared to the Smatch score, which calculates the F1 score, APE is a recall measure. The pyramid score is then computed the same way as discussed in Section 2: the total weight of all SCUs present in the candidate (w_p) divided by the total SCU weight possible for an average-length summary (w_{ideal}) (Nenkova and Passonneau, 2004). The APE algorithm[5] follows:

Algorithm 1 APE: computes the pyramid score for a given peer summary ($\boldsymbol{p}$) and SCUs ($\boldsymbol{c}$). w_{ideal} acts as a normalization factor.

```
function APE(p, c)
    w_p ← 0
    for all 1 ≤ i ≤ C do
        matched ← false
        d ← c_i
        for all 1 ≤ j ≤ D_i do
            for all 1 ≤ k ≤ P do
                if mmt(A_{d_j}, A_{p_k}) / |A_{d_j}| > τ then
                    w_p ← w_p + λ_i
                    matched ← true
                    break
            if matched then break
    return w_p / w_ideal
```

5 Results and Discussion

We selected TAC'09 corpus for evaluation, because 2009 was the last time DUC/TAC contained a general summarization task[6]. The corpus contains 44 topics. For each topic, there are 4 model summaries and the manually created pyramid. 55 participating systems were ranked according to the manual Pyramid scores, producing a gold ranking of the systems for each topic. Overall, there are

[4] Available at http://github.com/jflanigan/jamr.

[5] Altough the complexity of the algorithm is quite large, the runtime was not an issue in our experiments.

[6] In 2010 and 2011, TAC shifted towards aspect-based summarization. We also consider only initial summaries. Evaluating update summaries would need some changes similarly to the ROUGE adaptation in (Conroy et al., 2011).

$44 \times 55 = 2420$ evaluation scores. An automatic evaluation metric runs 2420 evaluations as well, and we can study its correlation with the manual pyramid score (the *per-summary* scenario). The other option is to calculate an average score for each system and look at the correlation of the averages (55 scores): the *per-system* scenario. The TAC'09 AESOP task concluded that on the system level, the correlation is sufficient (the best Pearson's r was .978, ROUGE-SU4: .921), but the metrics are much worse on the summary level.

ROUGE-SU4[7] and BE-HM were taken as baselines. We report three types of correlations. Pearson's tests the strength of a linear dependency between the two sequences. Spearman rank correlation is preferable for ordinal comparisons, absolute values are less relevant. Kendall's tau is less sensitive to outliers. Results are in Table 1.

We can observe that ROUGE-SU4 correlates better than BE-HM and about the same as APE (.66 Pearson, .62 Spearman, .45 Kendall). These numbers do not show whether the information added by APE's relations improves the summary evaluation. However, if we combine APE and ROUGE-SU4 by a linear combination ($\alpha \times$ APE $+ (1 - \alpha) \times$ ROUGE-SU4), we see a significant improvement. In the case of a "blind combination" (i.e. without looking at the test data, $\alpha = .5$), we notice an absolute improvement of 3.7% in Pearson, 4.0% in Spearman, and 3.2% in Kendall. In the case of the optimal $\alpha = .2$, a lower weight of the AMR relations, we see an improvement of 5.1% in Pearson, 6.4% in Spearman, and 5.5% in Kendall. The improvement is much larger than if BE-HM is combined with ROUGE-SU4. Combining all the three metrics yields only a marginal improvement. Figure 2 shows how the correlation depends on α.

We also noticed a large correlation between the gold number of SCUs (annotated in peers) and APE's number of SCUs: Pearson's r was .769. This further shows the positive effect of AMR relations.

The optimal setting for τ was .75, we tested all levels of τ with step .05. This means that 3/4 of the triples has to match between an SCU mention and a peer sentence to consider the SCU captured.

We further studied how different relation types affect the metric. The JAMR parser found 57 relation types. 51 relation types affected the correlation positively. These included verb arguments,

[7]ROUGE-SU4 was selected to represent the ROUGE family because it correlated best in the TAC AESOP experiments.

Metric	α	Pearson	Spearman	Kendall
ROUGE-SU4		.662	.624	.453
BE-HM		.612	.579	.413
APE		.661	.623	.456
APE + ROUGE-SU4	.2	**.713**	**.688**	**.508**
APE + ROUGE-SU4	.5	.695	.664	.485
BE-HM + ROUGE-SU4	.2	.666	.633	.460
BE-HM + ROUGE-SU4	.5	.661	.634	.461

Table 1: Correlations of the evaluation metrics with the manual pyramid score. Except for ROUGE-SU4 vs. APE, all differences are statistically significant with p-value=.01 measured by t-test and Fisher's transformation.

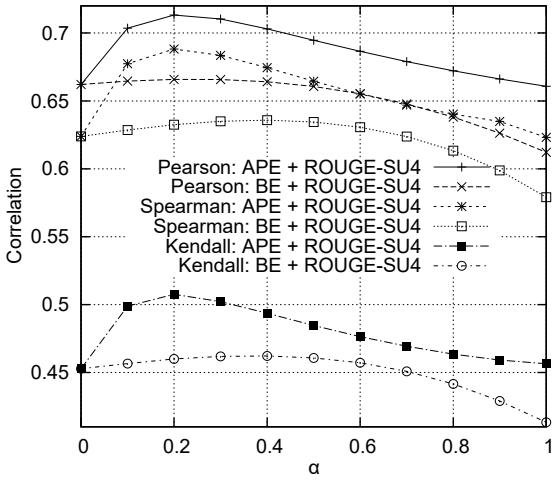

Figure 2: Linear combination of the metrics compared with human scores.

operands[8] (relations linking entity name parts), number-based relations (e.g. *unit*), time-based (e.g. date parts), prepositions, and most of the semantic relations, including the *polarity*. Six relation types had a negative effect. The *name* relation probably boosted the weight of the named entities too much. Syntactically oriented relations (e.g. *mod*) do not seem to be useful.

6 Conclusion

We showed that semantic relations among sentence items can improve the current summarization evaluation metrics. The approach is very sensitive to the quality of the AMR parser. It is expected to improve when AMR parsing advances. Our future work includes experiments with the latest findings in AMR parsing, e.g. Wang et al. (2015); Damonte et al. (2016). There has been made much progress on the Semantic Textual Similarity (STS) shared task in recent years (Agirre et al., 2016). We plan

[8]The opX relations were merged into a common "op" type.

to incorporate the metrics to further boost the evaluation performance. We found optimal settings of the parameters (τ and α), however, we need to investigate whether it generalizes across all summarization domains. As SCUs need to be manually created before performing summary evaluation, our final future work includes finding a way to use AMR to extract SCUs from model summaries and create the pyramid.

Acknowledgements

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports under the program NPU I., by Grant No. SGS-2016-018 Data and Software Engineering for Advanced Applications and by project MediaGist, EUs FP7 People Programme (Marie Curie Actions), no. 630786.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 497–511. http://www.aclweb.org/anthology/S16-1081.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Association for Computational Linguistics, Sofia, Bulgaria, pages 178–186. http://www.aclweb.org/anthology/W13-2322.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 748–752. http://www.aclweb.org/anthology/P13-2131.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2011. Nouveau-rouge: A novelty metric for update summarization. *Computational Linguistics* 37:1–8.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2016. An incremental parser for abstract meaning representation. *arXiv preprint at arXiv:1608.06111* .

Hoa T. Dang and Karolina Owczarzak. 2009. Overview of the tac 2009 summarization track. In National Institute of Standards and Technology, editors, *Proceedings of Text Analysis Conference (TAC-09)*. Gaithersburg, MD.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 1426–1436. http://www.aclweb.org/anthology/P14-1134.

George Giannakopoulos and Vangelis Karkaletsis. 2011. Autosummeng and memog in evaluating guided summaries. In National Institute of Standards and Technology, editors, *Proceedings of Text Analysis Conference (TAC-11)*. Gaithersburg, MD.

Aaron Harnly, , Ani Nenkova, Rebecca Passonneau, and Owen Rambow. 2005. Automation of summary evaluation by the pyramid method. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*. Borovets, Bulgaria.

Eduard Hovy, Chin-Yew Lin, and Liang Zhou. 2005. Evaluating duc 2005 using basic elements. In *Proceedings of the Fifth Document Understanding Conference (DUC)*. Vancouver, Canada.

Chin-Yew Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*. Barcelona, Spain.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of HLT-NAACL*. Edmonton, Canada.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Abstract meaning representation for sembanking. In *Proceedings of the 2015 Annual Conference of the North American Chapter of the ACL*. Association for Computational Linguistics, Denver, Colorado, page 10771086.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Vancouver, British Columbia, Canada, pages 523–530. http://www.aclweb.org/anthology/H/H05/H05-1066.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Meeting of the*

North American Chapter of the Association for Computational Linguistics (NAACL).

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing* 4(2).

Paul Over, Hoa Dang, and Donna Harman. 2007. DUC in context. *Information Processing and Management* 43(6):1506–1520. Special Issue on Text Summarisation (Donna Harman, ed.).

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*. Association for Computational Linguistics, Montréal, Canada, pages 1–9. http://www.aclweb.org/anthology/W12-2601.

Karolina Owczarzak and Hoa T. Dang. 2010. Overview of the tac 2010 summarization track. In National Institute of Standards and Technology, editors, *Proceedings of Text Analysis Conference (TAC-10)*. Gaithersburg, MD.

Karolina Owczarzak and Hoa T. Dang. 2011. Overview of the tac 2011 summarization track: Guided task and aesop task. In National Institute of Standards and Technology, editors, *Proceedings of the Text Analysis Conference (TAC-11)*. Gaithersburg, MD.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics* 31(1):71–106.

Rebecca J. Passonneau, Emily Chen, Weiwei Guo, and Dolores Perin. 2013. Automated pyramid scoring of summaries using distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sofia, Bulgaria, page 143147.

Stephen Tratz and Eduard Hovy. 2008. Summarization evaluation using transformed basic elements. In National Institute of Standards and Technology, editors, *Proceedings of Text Analysis Conference (TAC-08)*. Gaithersburg, MD.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. Boosting transition-based amr parsing with refined actions and auxiliary analyzers. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, pages 857–862. http://www.aclweb.org/anthology/P15-2141.