Wrapping up a Summary: from Representation to Generation

Josef Steinberger and **Marco Turchi** and
Mijail Kabadjov and **Ralf Steinberger**
EC Joint Research Centre
21027, Ispra (VA), Italy
{Josef.Steinberger, Marco.Turchi,
Mijail.Kabadjov, Ralf.Steinberger}
@jrc.ec.europa.eu

Nello Cristianini
University of Bristol,
Bristol, BS8 1UB, UK
nello@support-vector.net

Abstract

The main focus of this work is to investigate robust ways for generating summaries from summary representations without recurring to simple sentence extraction and aiming at more human-like summaries. This is motivated by empirical evidence from TAC 2009 data showing that human summaries contain on average more and shorter sentences than the system summaries. We report encouraging preliminary results comparable to those attained by participating systems at TAC 2009.

1 Introduction

In this paper we adopt the general framework for summarization put forward by Spärck-Jones (1999) – which views summarization as a threefold process: interpretation, transformation and generation – and attempt to provide a clean instantiation for each processing phase, with a particular emphasis on the last, summary-generation phase often omitted or over-simplified in the mainstream work on summarization.

The advantages of looking at the summarization problem in terms of distinct processing phases are numerous. It not only serves as a common ground for comparing different systems and understanding better the underlying logic and assumptions, but it also provides a neat framework for developing systems based on clean and extendable designs. For instance, Gong and Liu (2002) proposed a method based on Latent Semantic Analysis (LSA) and later J. Steinberger et al. (2007) showed that solely by enhancing the first source *interpretation* phase, one is already able to produce better summaries.

There has been limited work on the last summary generation phase due to the fact that it is unarguably a very challenging problem. The vast amount of approaches assume simple sentence selection, a type of extractive summarization, where often the summary representation and the end summary are, indeed, conflated.

The main focus of this work is, thus, to investigate robust ways for generating summaries from summary representations without recurring to simple sentence extraction and aiming at more human-like summaries. This decision is also motivated by empirical evidence from TAC 2009 data (see table 1) showing that human summaries contain on average more and shorter sentences than the system summaries. The intuition behind this is that, by containing more sentences, a summary is able to capture more of the important content from the source.

Our initial experimental results show that our approach is feasible, since it produces summaries, which when evaluated against the TAC 2009 data[1] yield ROUGE scores (Lin and Hovy, 2003) comparable to the participating systems in the Summarization task at TAC 2009. Taking into account that our approach is completely unsupervised and language-independent, we find our preliminary results encouraging.

The remainder of the paper is organised as follows: in the next section we briefly survey the related work, in §3 we describe our approach to summarization, in §4 we explain how we tackle the generation step, in §5 we present and discuss our experimental results and towards the end we conclude and give pointers to future work.

2 Related Work

There is a large body of literature on summarization (Hovy, 2005; Erkan and Radev, 2004; Kupiec et al., 1995). The most closely related work to the approach presented hereby is work on summarization attempting to go beyond simple sentence ex-

[1] http://www.nist.gov/tac/

Proceedings of the ACL 2010 Conference Short Papers, pages 382–386,
Uppsala, Sweden, 11-16 July 2010.

traction and to a lesser degree work on sentence compression. We survey below work along these lines.

Although our approach is related to sentence compression (Knight and Marcu, 2002; Clarke and Lapata, 2008), it is subtly different. Firstly, we reduce the number of terms to be used in the summary at a global level, not at a local per-sentence level. Secondly, we directly exploit the resulting structures from the SVD making the last generation step fully aware of previous processing stages, as opposed to tackling the problem of sentence compression in isolation.

A similar approach to our sentence reconstruction method has been developed by Quirk et al. (2004) for paraphrase generation. In their work, training and test sets contain sentence pairs that are composed of two different proper English sentences and a paraphrase of a source sentence is generated by finding the optimal path through a paraphrases lattice.

Finally, it is worth mentioning that we are aware of the 'capsule overview' summaries proposed by Boguraev and Kennedy (1997) which is similar to our TSR (see below), however, as opposed to their emphasis on a suitable browsing interface rather than producing a readable summary, we precisely attempt the latter.

3 Three-fold Summarization: Interpretation, Transformation and Generation

We chose the LSA paradigm for summarization, since it provides a clear and direct instantiation of Spärck-Jones' three-stage framework.

In LSA-based summarization the interpretation phase takes the form of building a term-by-sentence matrix $A = [A_1, A_2, \ldots, A_n]$, where each column $A_j = [a_{1j}, a_{2j}, \ldots, a_{nj}]^T$ represents the weighted term-frequency vector of sentence j in a given set of documents. We adopt the same weighting scheme as the one described in (Steinberger et al., 2007), as well as their more general definition of term entailing not only unigrams and bigrams, but also named entities.

The transformation phase is done by applying singular value decomposition (SVD) to the initial term-by-sentence matrix defined as $A = U\Sigma V^T$.

The generation phase is where our main contribution comes in. At this point we depart from standard LSA-based approaches and aim at producing a succinct summary representation comprised only of salient terms – Term Summary Representation (TSR). Then this TSR is passed on to another module which attempts to produce complete sentences. The module for sentence reconstruction is described in detail in section 4, in what follows we explain the method for producing a TSR.

3.1 Term Summary Representation

To explain how a term summary representation (TSR) is produced, we first need to define two concepts: *salience score* of a given term and *salience threshold*. Salience score for each term in matrix A is given by the magnitude of the corresponding vector in the matrix resulting from the dot product of the matrix of left singular vectors with the diagonal matrix of singular values. More formally, let $T = U \cdot \Sigma$ and then for each term i, the salience score is given by $|\vec{T_i}|$. Salience threshold is equal to the salience score of the top k^{th} term, when all terms are sorted in descending order on the basis of their salience scores and a cutoff is defined as a percentage (e.g., top 15%). In other words, if the total number of terms is n, then $100 * k/n$ must be equal to the percentage cutoff specified.

The generation of a TSR is performed in two steps. First, an initial pool of sentences is selected by using the same technique as in (Steinberger and Ježek, 2009) which exploits the dot product of the diagonal matrix of singular values with the right singular vectors: $\Sigma \cdot V^T$.[2] This initial pool of sentences is the output of standard LSA approaches.

Second, the terms from the source matrix A are identified in the initial pool of sentences and those terms whose *salience score* is above the *salience threshold* are copied across to the TSR. Thus, the TSR is formed by the most (globally) salient terms from each one of the sentences. For example:

- Extracted Sentence: "Irish Prime Minister Bertie Ahern admitted on Tuesday that he had held a series of private one-on-one meetings on the Northern Ireland peace process with Sinn Fein leader Gerry Adams, but denied they had been secret in any way."
- TSR Sentence at 10%: "Irish Prime Minister Bertie Ahern Tuesday had held one-on-one meetings Northern Ireland peace process Sinn Fein leader Gerry Adams"[3]

[2] Due to space constraints, full details on that step are omitted here, see (Steinberger and Ježek, 2009).

[3] The TSR sentence is stemmed just before feeding it to the reconstruction module discussed in the next section.

Average number of:	Human Summaries	System Summaries	At 100%	At 15%	At 10%	At 5%	At 1%
Sentences/summary	6.17	3.82	3.8	3.95	4.39	5.18	12.58
Words/sentence	15.96	25.01	26.24	25.1	22.61	19.08	7.55
Words/summary	98.46	95.59	99.59	99.25	99.18	98.86	94.96

Table 1: Summary statistics on TAC'09 data (initial summaries).

Metric	$\text{LSA}_{extract}$	At 100%	At 15%	At 10%	At 5%	At 1%
ROUGE-1	0.371	0.361	0.362	0.365	0.372	0.298
ROUGE-2	0.096	0.08	0.081	0.083	0.083	0.083
ROUGE-SU4	0.131	0.125	0.126	0.128	0.131	0.104

Table 2: Summarization results on TAC'09 data (initial summaries).

4 Noisy-channel model for sentence reconstruction

This section describes a probabilistic approach to the reconstruction problem. We adopt the noisy-channel framework that has been widely used in a number of other NLP applications. Our interpretation of the noisy channel consists of looking at a stemmed string without stopwords and imagining that it was originally a long string and that someone removed or stemmed some text from it. In our framework, reconstruction consists of identifying the original long string.

To model our interpretation of the noisy channel, we make use of one of the most popular classes of SMT systems: the Phrase Based Model (PBM) (Zens et al., 2002; Och and Ney, 2001; Koehn et al., 2003). It is an extension of the noisy-channel model and was introduced by Brown et al. (1994), using phrases rather than words. In PBM, a source sentence f is segmented into a sequence of I phrases $f^I = [f_1, f_2, \dots f_I]$ and the same is done for the target sentence e, where the notion of phrase is not related to any grammatical assumption; a phrase is an n-gram. The best translation e_{best} of f is obtained by:

$$e_{best} = arg\max_e p(e|f) = arg\max_e \prod_{i=1}^{I} \phi(f_i|e_i)^{\lambda_\phi}$$
$$d(a_i - b_{i-1})^{\lambda_d} \prod_{i=1}^{|e|} p_{LM}(e_i|e_1 \dots e_{i-1})^{\lambda_{LM}}$$

where $\phi(f_i|e_i)$ is the probability of translating a phrase e_i into a phrase f_i. $d(a_i - b_{i-1})$ is the distance-based reordering model that drives the system to penalize substantial reorderings of words during translation, while still allowing some flexibility. In the reordering model, a_i denotes the start position of the source phrase that was translated into the i^{th} target phrase, and b_{i-1} denotes the end position of the source phrase translated into the $(i-1^{\text{th}})$ target phrase. $p_{LM}(e_i|e_1 \dots e_{i-1})$ is the language model probability that is based on the Markov chain assumption. It assigns a higher probability to fluent/grammatical sentences. λ_ϕ, λ_{LM} and λ_d are used to give a different weight to each element (for more details see (Koehn et al., 2003)).

In our reconstruction problem, the difference between the source and target sentences is not in terms of languages, but in terms of forms. In fact, our source sentence f is a stemmed sentence without stopwords, while the target sentence e is a complete English sentence. "Translate" means to reconstruct the most probable sentence e given f inserting new words and reproducing the inflected surface forms of the source words.

4.1 Training of the model

In Statistical Machine Translation, a PBM system is trained using parallel sentences, where each sentence in a language is paired with another sentence in a different language and one is the translation of the other.

In the reconstruction problem, we use a set, S_1 of 2,487,414 English sentences extracted from the news. This set is duplicated, S_2, and for each sentence in S_2, stopwords are removed and the remaining words are stemmed using Porter's stemmer (Porter, 1980). Our stopword list contains 488 words. Verbs are not included in this list, because they are relevant for the reconstruction task. To optimize the lambda parameters, we select 2,000 pairs as development set.

An example of training sentence pair is:

- Source Sentence: "royal mail ha doubl profit 321 million huge fall number letter post"
- Target Sentence: "royal mail has doubled its profits to 321 million despite a huge fall in the number of letters being posted"

In this work we use Moses (Koehn et al., 2007), a complete phrase-based translation toolkit for academic purposes. It provides all the state-of-the-art components needed to create a phrase-based machine translation system. It contains different modules to preprocess data, train the Language Models and the Translation Models.

5 Experimental Results

For our experiments we made use of the TAC 2009 data which conveniently contains human-produced summaries against which we could evaluate the output of our system (NIST, 2009).

To begin our inquiry we carried out a phase of exploratory data analysis, in which we measured the average number of sentences per summary, words per sentence and words per summary in human vs. system summaries in the TAC 2009 data. Additionally, we also measured these statistics of summaries produced by our system at five different percentage cutoffs: 100%, 15%, 10%, 5% and 1%. [4] The results from this exploration are summarised in table 1. The most notable thing is that human summaries contain on average more and shorter sentences than the system summaries (see 2nd and 3rd column from left to right). Secondly, we note that as the percentage cutoff decreases (from 4th column rightwards) the characteristics of the summaries produced by our system are increasingly more similar to those of the human summaries. In other words, within the 100-word window imposed by the TAC guidelines, our system is able to fit more (and hence shorter) sentences as we decrease the percentage cutoff.

Summarization performance results are shown in table 2. We used the standard ROUGE evaluation (Lin and Hovy, 2003) which has been also used for TAC. We include the usual ROUGE metrics: R_1 is the maximum number of co-occurring unigrams, R_2 is the maximum number of co-occurring bigrams and R_{SU4} is the skip bigram measure with the addition of unigrams as counting unit. The last five columns of table 2 (from left to right) correspond to summaries produced by our system at various percentage cutoffs. The 2nd column, $LSA_{extract}$, corresponds to the performance of our system at producing summaries by sentence extraction only.[5]

In the light of the above, the decrease in performance from column $LSA_{extract}$ to column 'At 100%' can be regarded as reconstruction error.[6] Then, as we decrease the percentage cutoff (from 4th column rightwards) we are increasingly covering more of the content comprised by the human summaries (as far as the ROUGE metrics are able to gauge this, of course). In other words, the improvement of content coverage makes up for the reconstruction error, and at 5% cutoff we already obtain ROUGE scores comparable to $LSA_{extract}$. This suggests that if we improve the quality of our sentence reconstruction we would potentially end up with a better performing system than a typical LSA system based on sentence selection. Hence, we find these results very encouraging.

Finally, we admittedly note that by applying a percentage cutoff on the initial term set and further performing the sentence reconstruction we gain in content coverage, to a certain extent, on the expense of sentence readability.

6 Conclusion

In this paper we proposed a novel approach to summary generation from summary representation based on the LSA summarization framework and on a machine-translation-inspired technique for sentence reconstruction.

Our preliminary results show that our approach is feasible, since it produces summaries which resemble better human summaries in terms of the average number of sentences per summary and yield ROUGE scores comparable to the participating systems in the Summarization task at TAC 2009. Bearing in mind that our approach is completely unsupervised and language-independent, we find our results promising.

In future work we plan on working towards improving the quality of our sentence reconstruction step in order to produce better and more readable sentences.

[4] Recall from section §3 that the salience threshold is a function of the percentage cutoff.

[5] These are, effectively, what we called initial pool of sentences in section 3, before the TSR generation.

[6] The only difference between the two types of summaries is the reconstruction step, since we are including 100% of the terms.

References

B. Boguraev and C. Kennedy. 1997. Salience-based content characterisation of text documents. In I. Mani, editor, *Proceedings of the Workshop on Intelligent and Scalable Text Summarization at the Annual Joint Meeting of the ACL/EACL*, Madrid.

P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1994. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

J. Clarke and M. Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:273–318.

G. Erkan and D. Radev. 2004. LexRank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*.

Y. Gong and X. Liu. 2002. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of ACM SIGIR*, New Orleans, US.

E. Hovy. 2005. Automated text summarization. In Ruslan Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, pages 583–598. Oxford University Press, Oxford, UK.

K. Knight and D. Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

P. Koehn, F. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL '03*, pages 48–54, Morristown, NJ, USA.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL '07, demonstration session.*

J. Kupiec, J. Pedersen, and F. Chen. 1995. A trainable document summarizer. In *Proceedings of the ACM SIGIR*, pages 68–73, Seattle, Washington.

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

NIST, editor. 2009. *Proceeding of the Text Analysis Conference*, Gaithersburg, MD, November.

F. Och and H. Ney. 2001. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL '02*, pages 295–302, Morristown, NJ, USA.

M. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

C. Quirk, C. Brockett, and W. Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP*, volume 149. Barcelona, Spain.

K. Spärck-Jones. 1999. Automatic summarising: Factors and directions. In I. Mani and M. Maybury, editors, *Advances in Automatic Text Summarization*. MIT Press.

J. Steinberger and K. Ježek. 2009. Update summarization based on novel topic distribution. In *Proceedings of the 9th ACM DocEng, Munich, Germany*.

J. Steinberger, M. Poesio, M. Kabadjov, and K. Ježek. 2007. Two uses of anaphora resolution in summarization. *Information Processing and Management*, 43(6):1663–1680. Special Issue on Text Summarisation (Donna Harman, ed.).

R. Zens, F. J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *Proceedings of KI '02*, pages 18–32, London, UK. Springer-Verlag.