

An Evolutionary Algorithm for Automatic Summarization

Aurélien Bossard* and Christophe Rodrigues**

*LIASD - EA 4383, Université Paris 8, Saint-Denis, France
bossard@iut.univ-paris8.fr

**Léonard de Vinci Pôle Universitaire, Research Center, Paris La Défense, France
christophe.rodrigues.bento@gmail.com

Abstract

This paper proposes a novel method to select sentences for automatic summarization based on an evolutionary algorithm. The algorithm explores candidate summaries space following an objective function computed over ngrams probability distributions of the candidate summary and the source documents. This method does not consider a summary as a stack of independent sentences but as a whole text, and makes use of advances in unsupervised summarization evaluation. We compare this sentence extraction method to one of the best existing methods which is based on integer linear programming, and show its efficiency on three different acknowledged corpora.

1 Introduction

Automatic summarization systems are essential components of information systems. Indeed, increase of numerical information sources can have a negative effect on online content reading and assimilation. Summarizing such content can allow users to better apprehend it. Automatic summarization has therefore become one of the first research in natural language processing field (Luhn, 1958) and still remains a widely spread topic.

In order to validate the benefits obtained from new methods or parametrization, the automatic summarization field needs robust evaluation methods. Evaluating automatic summaries, just like evaluating automatic translation, is a complex task. Until early 2000s, only two types of approach existed: entirely manual evaluation with a reading grid and semi-automatic evaluations that compare automatic summaries with human written references. Since, entirely automatic approaches

that allow for evaluating a summary without a human reference (writing it is the most time-consuming task in evaluation) emerged and have recently achieved good performances, using probabilistic models (Louis and Nenkova, 2009; Sagion et al., 2010).

Probabilistic models for automatic summarization evaluation are natural: a summary and its source have to share the same distribution of concepts. As they have proven performant for evaluation, using them to guide automatic summarization process seems obvious, although it has to our knowledge not been already tested. As opposed to the most part of automatic summarization methods that use encoded metrics, we here propose to consider automatic summarization as the maximization of a natural score: the divergence between the concepts distribution in the source and the concepts distribution of a candidate summary.

So we view automatic summarization as choosing the best summary among a very large set of candidate summaries upon a metric that is computed on the whole candidate summary. This leads us to the use of an evolutionary algorithm in order to naturally select the best summaries.

While other recent papers (Li et al., 2013; Nishikawa et al., 2014; Peyrard and Eckle-Kohler, 2016) integrate sophisticated and task-specific preprocessings and postprocessings, and handle semantics using complex representations, this paper proposes a new generic and directly usable sentence extraction method for automatic summarization. This method explores the candidate summaries space using an evolutionary algorithm. This algorithm aims at finding an approximate solution of the maximization of an objective function computed over a candidate summary. We first present iterative methods and exploratory analysis methods for automatic summarization. We then expose our sentence extraction method and

the evaluation protocol: it is compared to one of the best methods in the automatic summarization field. Finally, we discuss our results.

2 Related Work

Iterative Selection Automatic summarization systems generally combine a centrality score for text portions and an extraction method for these portions. The first automatic summarization systems (Luhn, 1958; Edmundson, 1969) simply extracted most central portions. MMR method (Carbonell and Goldstein, 1998) allows for iterative text portions extraction given a centrality score and a redundancy score. CSIS-based method (Radev, 2000) removes from a list of text portions sorted by centrality every one that shares too much information with a higher ranked. These methods share a major drawback: generated summaries depend mostly on the first selected text portion. Therefore, they are exposed to omitting summaries made of average ranked sentences that reflect correctly the overall content of the source documents when combined together.

Optimization Other methods emerged recently to overcome this problem. They consist in exploring the space of all candidate summaries in order to find the one that maximizes an objective function. This problem is exponential in input sentences as there are C_m^n candidate summaries composed of n sentences for a corpus of m sentences. For example, choosing 10 sentences over 200 leads to 10^{25} possible solutions. Adding constraints on text portions selection and using ILP¹ help delimiting the problem and finding (not always) an exact solution (McDonald, 2007; Gillick and Favre, 2009b). The search space is limited by constraints on text portions length and by constraints that avoid including text portions that do not provide additional information. Whereas Gillick and Favre (2009b) select summaries based on the maximization of bigram occurrences, Li et al. (2013) try to maximize the similarity between summaries and sources bigram frequencies. These methods have proven to be very efficient. However, the use of an ILP solver enforces to modelize the problem as a linear functions, so does not allow for complex functions that could better take into account the structure of the automatic summarization problem.

¹Integer Linear Programming

Liu et al. (2006); Nandhini and Balasundaram (2013); Shigematsu and Kobayashi (2014) propose to look for an approximate solution using a genetic algorithm. As opposed to ILP extraction methods, these methods are free of any constraints on the objective function. However, these methods keep on considering a summary as a set of independent portions of text, and do not take advantage of the new structure of the problem that we propose: a summary is not considered as a whole. Considering a summary as a whole allows for a better space exploration and more complex objective functions. Alfonseca and Rodríguez (2003) use a "standard GA" without describing it and several fitness scores, whose main metric is cosine-tfidf similarity between sources and candidate summaries. However, this similarity has obtained poor results when used as an automatic evaluation metric (Nenkova et al., 2007). So this metric should not be used as fitness score for automatic summarization. As opposed to Alfonseca and Rodríguez (2003), we define a new, non-standard, and fully replicable evolutionary algorithm combined with an extension of an agreed-upon automatic evaluation metric.

Supervised Learning (Litvak et al., 2010; Bossard and Rodrigues, 2011) use genetic algorithms for supervised learning of parameters in order to tune automatic summarization systems. Nishikawa et al. (2014); Takamura and Okumura (2010); Sipos et al. (2012) perform structured output learning to maximize ROUGE scores. These approaches suffer from the complexity of machine learning model. Moreover, it requires learning data and is very task-specific. Recently, Peyrard and Eckle-Kohler (2016) proposed to use an approximation of ROUGE-N score combined to an ILP solver. The approximation of ROUGE-N score is performed with supervised learning. The results obtained outperform state-of-the-art methods on DUC 2002 and 2003 corpora.

In contrast to supervised learning, we propose to use a fully unsupervised method that allows for summarizing even when no manual reference is available. We here propose to use scoring functions based on probabilistic models of source documents and candidate summaries. The smoothing used for building probability distributions considers a candidate summary as a whole, not as a set of independent sentences. This complex objective function requires a constraint-free optimiza-

tion algorithm. So evolutionary algorithms seem appropriate. In the meantime, the scoring functions we use allow to benefit more of evolutionary algorithms.

3 Our Method

Louis and Nenkova (2009) have proposed an entirely unsupervised function for automatic summary evaluation. The evaluation method has proved efficient as it strongly correlates to Pyramid score, a semi-automatic score used in TAC evaluation campaigns to assess automatic summaries informational quality (Nenkova et al., 2007). Saggion et al. (2010) confirmed the relevance of the approach. The method is entirely unsupervised, so it does not need any human reference. This makes it perfectly fit to be used as an objective function in a guided space exploring algorithm.

Our objective function computes the probability distribution of tokens in the source documents with the probability distribution of tokens in the summaries. Tokens can be words, ngrams or even semantic concepts. The objective function is based on (Louis and Nenkova, 2009) that handles automatic evaluation using Jensen-Shannon(JS) divergence.

The JS divergence can be considered as a symmetric version of Kullback-Leibler divergence (KL). KL divergence is defined as the average number of bits wasted by coding samples belonging to P using another distribution Q , an approximate of P (Louis and Nenkova, 2009). In our case, the two distributions are those for words (or concepts) in the input and the summary.

Given two distributions P and Q , here is the definition of JS divergence:

$$JS(P||Q) = \frac{1}{2}[KL(P||A) + KL(Q||A)]$$

with : $A = \frac{P+Q}{2}$ the mean distribution of P and Q ; $KL(P||A)$ the Kullback-Leibler divergence:

$$KL(P||Q) = \sum_w p_P(w) \log_2 \frac{p_P(w)}{p_Q(w)}$$

Louis and Nenkova (2009) use a simple weighted Laplace smoothing over probabilities.

$$p(w) = \frac{C(w)+\delta}{N+\delta \times 1.5 \times |V|}$$

with : $C(w)$ the number of occurrences of w ; N the overall number of tokens; V the vocabulary, and $\delta = 0.0005$.

3.1 Unigram Distribution Objective Function

This objective function (Uniprob) fits the exact automatic evaluation function described above and in (Louis and Nenkova, 2009).

3.2 Bigram Simple Sum Objective Function

Bigram simple sum objective function (Bisimple) consists in summing all bigram weights in a candidate summary. Candidate summaries get a high score if they are composed of the most frequent bigrams in the source documents.

3.3 Bigram Cosine Objective Function

Bigram cosine objective function (Bicos) consists in a cosine similarity between source and candidate summaries bigram vector. This objective function is used in (Alfonseca and Rodríguez, 2003).

3.4 Bigram Distribution Objective Function

Lin (2004) has shown that ROUGE semi-automatic evaluation metrics are more correlated to manual evaluation when using bigrams rather than unigrams when it comes to compare summaries and references of a standard length: 50 words and more. Therefore, we make the assumption that a probabilistic model based on bigrams outcomes one based on unigrams for our objective function. Moreover, we smooth probabilities using Dirichlet smoothing (MacKay and Peto, 1994), that adds to every count of a token in a summary its probability in the source documents. Dirichlet smoothing is more faithful to the data (Zhai and Lafferty, 2004) than Laplacian smoothing while remaining simple to compute. With Dirichlet smoothing, the probability of a token t in a summary S is computed this way:

$$p_{dir}(t|S) = \frac{C_S(t) + \mu p_{ML}(t|D)}{N_S + \mu}$$

with: D the source documents; $C_S(t)$ the number of occurrences of t in S ; $p_{ML}(t|D)$ the maximum likelihood of t in D ; N_S the number of tokens in S ; and μ a constant parameter called pseudo-frequency.

Candidate summaries are subsets of source documents, so smoothing is only applied to summaries.

However, this objective function (Biprob) may have a major drawback: it favors summaries whose probability distributions are close to source documents. If these source documents are highly

redundant, the summaries that are rated high on this objective function would also be redundant.

3.5 Evolutionary Algorithm

We here describe the evolutionary algorithm we developed and use for maximization of our objective function. This evolutionary algorithm differs from traditional genetic algorithms as described below, to better handle automatic summarization problem. In fact, automatic summarization tasks often bound the summaries length in words, not in number of sentences. So the number of sentences extracted in a summary depends on both maximum summary length and each sentence length. That is why we cannot use a standard problem model in which every candidate summary would be an individual with n genes, n being the number of sentences to extract. Having a specific representation of the individuals leads to defining new operators on these individuals. So the mutation and hybridization mutators differ from what we are used to because of the structure of our problem.

Individuals Definition Each individual (= a candidate summary) is defined by a set of chromosomes (= sentences). A chromosome codes for a sentence. The number of chromosomes in the set of an individual is variable, while there is an upper threshold for the sum over the length of all chromosomes in words. This requires some adaptations to the classic mutation and hybridization operators, that we define below.

Algorithm Sequence At generation 0, a starting population is created randomly. It contains N individuals, where $N = N_p + N_m + N_h$, where N_p , N_m and N_h are the parameters of the evolutionary algorithm for the next generations. For each of the next generations:

- N_p is the number of parents;
- N_m is the number of mutated individuals;
- N_h is the number of hybridated individuals.

Then N_p parents are selected to generate by mutation and by hybridation N_m and N_h additional individuals. The parents and the individuals they generated constitute a new generation. N_g generations are iteratively created this way. At the end, the individual that maximizes the objective function is selected.

Starting Population Selection N individuals are randomly generated. A new sentence is randomly added to an individual until the length constraint is reached:

$$\exists s \in S \setminus I \text{ s.t. } \sum_{s_i \in I} \text{length}(s_i) + \text{length}(s) < \text{maxLength},$$

with I an individual and S the sentences in source documents.

Parents Selection Parents selection can be performed through different methods. These methods favor space exploration or exploitation by selecting the best individuals. We chose a selection method that is a compromise between exploration and exploitation : tournament selection. N_p tournaments composed of $\frac{N}{N_p}$ individuals – with N the total population size – are randomly created. The best individual in each tournament is selected as a parent for the next generation.

Mutation Operator We cannot use classic mutation operators: changing one gene for another. Selecting a too long sentence would violate the summary length constraint while selecting a too short sentence would let the new candidate summary to be evaluated against other summaries that are longer. Our mutation operator is defined as follows : a chromosome (sentence) is randomly deleted from an individual. The individual is then randomly filled with new chromosomes until the length constraint is no longer satisfiable.

Hybridization Operator The chromosomes of two individuals (here, parents) are put together in a single set. A new individual is then created with chromosomes randomly selected from this set. Once no more chromosome matches the length constraint, the individual is randomly filled with chromosomes from the source documents. This operation does not guarantee that chromosomes from both individuals will be selected, so it is completely different from traditional hybridization operators. Proceeding this way is however mandatory in order to create individuals that will satisfy the summary length constraint.

4 Experiment

We compare our summarization method to three *baselines* on TAC 2008, TAC 2009 evaluation campaigns corpora². We also use the French cor-

²The corpora are available on request at <http://www.nist.gov/tac/data/index.html>.

pus RPM2³ evaluation that share a similar structure with the TAC 2008 and 2009 corpora.

4.1 Corpora

TAC 2008, 2009 and RPM2 are composed of two distinct parts: the first one is dedicated to standard multi-document summarization. The second is dedicated to update summarization: the goal is to summarize information in the update set, assuming that the user has already read the information in the standard set. The following TAC campaigns concern guided summarization: writing summaries for a given topic where the topic falls into a predefined category, including "Accidents and Natural Disasters", "Attacks", "Health and Safety", "Endangered Resources", and "Trials and Investigations". The goal is to encourage a deeper linguistic analysis, and summarization systems must take into account each task specificity in order to produce coherent summaries. So, post 2009 TAC campaigns are out of this article scope.

TAC 2009 standard summarization corpus is composed of 44 sets of 10 documents each: english-written news articles. The task consists in generating a 100 words summary for every document set. Average document length is 6330 words.

4.2 Our System

Document Preprocessing The documents are first cleaned: all tags and meta-information are removed.

Tokenization and Sentence Splitting We first use the default tokenizer of the POS-tagging tool tree-tagger⁴ (Schmid, 1994). Tree-tagger is also used for POS-tagging (not used in the English system) and sentence splitting.

Stemming Words are stemmed via the Porter stemmer for TAC 2008 and 2009 corpora. For RPM2, we implemented the French snowball stemmer⁵.

Bigrams Filtering Bigrams composed of two stopwords are pruned. A stoplist composed of the 200 more frequent English words is used for this

purpose. For the French corpus, a word is considered as a stopword if it is tagged as either a determiner or a preposition.

Sentence Selection Sentence selection is performed by the evolutionary algorithm described in Section 3.5 using the three objective functions described in Sections 3.4, 3.1 and 3.2.

Evolutionary Algorithm Parameters We empirically tuned the number of generations using TAC 2009 test dataset in order to increase the probability that the algorithm converges. TAC 2009 test dataset and evaluation datasets are strictly disjoint. Here are the parameters of the evolutionary algorithm: $N_p = N_h = N_m = 160$ and $N_g=160$.

First Sentences Salience TAC 2009 corpus is a news corpus composed for the most part of news wires and articles. First sentences are often considered as more salient than the other ones. (Gillick et al., 2009a) upweight by a factor 2 the concepts that appear in the first sentence of a document and thus reach a 16% gain in automatic evaluation metrics on TAC 2009 corpus. Our systems also upweight by a factor 2 the bigrams in the first sentences. We test its effect in Section 5.

4.3 Baselines

We implemented two baselines:

- *lexmmr*: common scoring method LexRank (Erkan and Radev, 2004), followed by MMR (Carbonell and Goldstein, 1998) to perform redundancy removal;
- *ILP*: the sentence selection method of (Gillick et al., 2009a) described in Section 2.

These two systems have the same preprocessing, tokenizing, stemming, filtering, and first sentence bigrams weighting as ours, so they can be compared efficiently.

ILP1 baseline is based on ICSI summarization system (Gillick et al., 2009a) which is considered as state-of-the-art in a recent study (Hong et al., 2014). ICSI employ different heuristics/preprocessings, eg pronominal references removal, relative dates and "said" clauses removal. In order to efficiently compare sentence extraction methods, we here use a generic version of ICSI system: sentence extraction is processed using the

³The corpus is available on request at <http://lia.univ-avignon.fr/fileadmin/documents/rpm2/>

⁴Treetagger: <http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>

⁵Snowball: <http://snowball.tartarus.org/algorithms/french/stemmer.html>

		Baselines				Evolutionary algorithm based systems				
		lexmmr	hextac	ILP1	ILP2	uniprob	biprob	bisimple	bicos	Oracle
TAC2008	ROUGE-1	.3134	-	.3713	.3716	.3546	.3804	.3752	.3497	.4262
	ROUGE-2	.0647	-	.1075	.1027	.0816	.1103	.1065	.0930	.1718
TAC2009	ROUGE-1	.3361	.3794	.3729	.3837	.3509	.3855	.3846	.3586	.4348
	ROUGE-2	.0787	.1065	.1053	.1096	.0821	.1173	.1128	.0961	.1782
RPM2	ROUGE-1	.3203	-	.4126	.4126	.3843	.4250	.3801	.4071	.4407
	ROUGE-2	.0889	-	.1568	.1568	.1472	.1683	.1473	.1491	.2039
Overall	ROUGE-1	.3236	-	.3793	.3837	.3655	.3904	.3798	.3634	.4321
	ROUGE-2	.0745	-	.1154	.1151	.0961	.1234	.1163	.1042	.1800

Table 1: Average results of all systems on TAC 2008, 2009 and RPM2

same solver as ICSI system (glpk) but preprocessings are limited to the strict minimum: the same as our system, described in 4.2. This way it can be efficiently compared to our evolutionary based sentence extraction method. As in (Gillick et al., 2009a), sentences of less than 10 words and bigrams that do not appear more than twice are not taken into account; in ILP2, we add the last rule in ICSI summarization system: sentences that do not share at least one word in common with the query are pruned. This way, we can compare more efficiently ILP baselines to our system that does not take query into account. The only difference between ICSI summarization system and ILP2 lies in processings that are not related to the sentence selection module.

The last baseline (*hextac*) is TAC 2009 third *baseline* : human generated extractive summaries (Genest et al., 2009). This baseline stands for determining the best summary we can achieve using purely extractive methods. Therefore, comparing its results to a pure extractive summarizer is informative. However, as the goal of this system is to produce good summaries for a human judge, its automatic scores could be lower than manual ones. In fact, human extractors can emphasize linguistic quality and global coherence rather than pure information extraction.

4.4 Oracle Experiments

We want to estimate the upper bound that a system based on our evolutionary algorithm and a probabilistic model can achieve. For this purpose, we define a new objective function. The objective function described above compares distributions between candidate summaries and source documents. If we suppose that manual summaries are available (which is the case in ROUGE evaluation campaigns), then we can directly compute distributions between candidate summaries and manual ones. So doing, we obtain an evolutionary approach guided by an Oracle. Given the small size of manual summaries used as source documents,

we don't use any smoothing at all.

This oracle is a good way to fit to the manual summaries. However, it is not optimal in all cases. The manual summaries are indeed small and are not necessarily composed of bigrams also present in source documents. This can cause the distribution divergences to be less precise.

5 Results

We here use the same ROUGE parameters as for TAC evaluation campaigns⁶.

5.1 General Results

Table 1 presents the results obtained by all the systems and baselines described in Section 4.2 on three corpora : TAC 2008, TAC 2009 and RPM2. The best system, whatever the evaluation metric, is *biprob*, that uses our evolutionary algorithm and JS divergence between bigrams distributions on source documents and candidate summaries as objective function. It outperforms *ILP1* and *ILP2* baselines. *ILP1* and *ILP2* baselines have the same ROUGE scores on RPM2 corpus, as RPM2 is not query-oriented. One can notice that *biprob* and human baseline *hextac* are really close in terms of ROUGE scores.

Biprob system as well as *ILP* baselines outperform the human generated *hextac* baseline. It means that both systems succeed in extracting as many salient information as a human. However, *hextac* summaries outperform automatic summaries in terms of linguistic quality: coherence, ordering...

5.2 Impact of First Sentence Bigrams Upweight

All of the three corpora are only composed of newswire articles and share the same information structure. The central information is located in the first sentences. Figure 1 shows *biprob* and *ILP* baseline ROUGE-2 scores when the upweight

⁶ROUGE-1.5.5.pl -c 95 -r 1000 -n 2 -m -a -x -d file

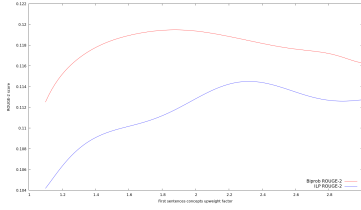


Figure 1: *Biprob* system and *ILP1* baseline results depending on first sentences concepts upweight value

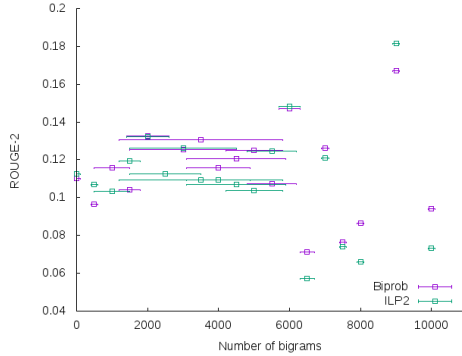


Figure 2: ROUGE-2 scores on all topics depending on topic size in bigrams

factor of first sentence bigrams varies on our three corpora. Increasing first sentences concepts weight improves ROUGE-2 scores of both ILP baseline and *biprob* on the overall score over the three corpora. The maximum is reached around the factor 2, as assumed by Gillick et al. (2009a). Our experiment confirms that weighting up first sentence concepts with a factor 2 is efficient on newswire articles corpora.

5.3 Impact of Input Size

Figure 2 presents ROUGE-2 scores of ILP1 baseline and *Biprob* depending on the number of bigrams in the source documents for all corpora. The x error bar shows the number of topics that are concerned by a point. One can see that the two systems are really close for small topics. However, for topics with a larger number of bigrams, *Biprob* tends to outperform ILP1 (except for number of bigrams intervals with a small number of examples).

5.4 Impact of Bigram Filtering

Taking into account only the bigrams that appear more than twice for ILP baselines increases results (as also noticed in (Gillick et al., 2009a)), so we tested a probability distribution that keeps

only these bigrams. The results are not as good as when computing probability distribution over all bigrams. This can be linked to the fact that our method performs better when the number of bigrams as input is high.

5.5 Convergence on TAC 2009 Data

Figure 3 shows the average fitness score (= bigram distribution objective function) of each generation for every topic of TAC 2008, TAC 2009 and RPM2 corpora for 12 different runs of the evolutionary algorithm. We can see that the scores seem to converge quite fast on average, before the 50th generation.

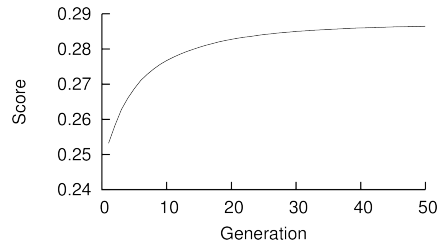


Figure 3: Average fitness scores of all topics in TAC 2008, 2009 and RPM2 corpora depending on the generation.

The Figure 3 presents the fitness average scores on all evaluation topics. Convergence speed depend on several factors: data dispersal, exploration space size... Figure 4 shows the convergence speed for the most populated topic of our three corpora: the D0918 topic from TAC 2009. It prints the average score, and the score standard deviation for D0918 topic on 12 different runs. This can give us an idea of the algorithm convergence in the worst case, speaking of exploration space size.

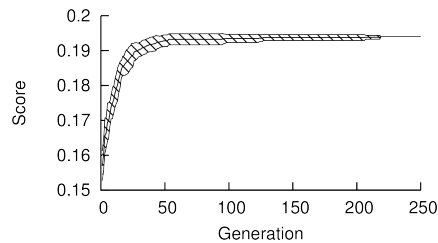


Figure 4: Average scores and standard deviation of scores for D0918 topic depending on the generation.

The algorithm converges near the 220th generation, and the topic is composed of 575 sentences. There are more than 5×10^{13} candidate summaries

that match the 100 words length limit, and the evolutionary algorithm has always reached convergence after exploring 70400 different candidate summaries.

5.6 Oracle Results

One can see in table 1 that, as expected, our Oracle performs more than 50% better than every of our systems. This means that there are summaries composed of sentences in the source documents that are closer to the reference summaries than those chosen by our objective functions. The 50% difference in ROUGE-2 scores shows that there is room for improvement.

6 Discussion

In this article, we proposed a sentence extraction method for automatic summarization that allows for good summaries on our three evaluation corpora. Our method outperforms a state-of-the-art sentence extraction method based on an ILP model by more than 6.4% on the combined score of our three evaluation corpora.

We show that Jensen-Shannon divergence outperforms cosine similarity as objective function in our evolutionary algorithm. This confirms the conclusions of (Louis and Nenkova, 2009) on the evaluation of metrics for automatic summary evaluation.

Thanks to the use of an evolutionary algorithm rather than an ILP-based method, the score computation and constraints are totally free and not limited by ILP modelization constraints. However, the objective functions described here do not handle redundancy. A candidate summary will be well scored if its probability distribution sticks to the one of the source documents. So, if the source documents display high redundancy on a given information, our method could generate redundant summaries. However, other objective functions that handle redundancy can be implemented as well as constraints on candidate summaries generation. Our system does not try to improve linguistic quality: sentences articulation is not managed. Methods such as lexical chains (Barzilay and Elhadad, 1999) could be implemented to the objective function or used as post-treatment.

The method proposed in the article uses an evolutionary algorithm and objective functions that imply heavy computation. Several solutions can be brought to this problem: using GPU architec-

ture for faster computation of distribution divergences as the evolutionary algorithm can be easily parallelized, or finding a linear approximation of the divergence distribution function so it can be used inside an ILP solver.

At last, an evolutionary algorithm needs a fine parameters tuning: population size, mutants and cross over percentage. Although we parameterized the algorithm so it finds a good solution for every topic of our three corpora, their influence on the algorithm convergence should be further studied using different data sets.

Extractive automatic summaries can outperform (in terms of ROUGE scores) human extractive summaries. It brings us to the upper limit that one can achieve with purely extractive methods, and to question the methods to consider from now on: sentence compression, generative paradigms for specialized summaries, or sentence rephrasing to achieve a better textual cohesion.

7 Conclusion

For automatic summarization, finding a summary (as good as possible) without an Oracle or any reference summary is an unsupervised task. In this paper, we show that automatic summarization can be achieved using recent advances in automatic summarization evaluation. In particular, we explore automatic summarization under the hypothesis that source documents and summaries have to share the same bigram distribution.

We present a sentence extraction evolutionary algorithm for automatic summarization capable of improving summary generation after generation based only on distribution computation. We show that this extraction algorithm achieves good results compared to state-of-the-art method on three corpora. However, the system can be improved in several ways: the objective function can be improved to better evaluate candidate summaries or be approximated to be used in a faster algorithm. If ROUGE-2 scores used in this paper are heavily correlated to human metrics, the evaluation needs to be pushed further using fully manual metrics such as Pyramid.

References

- Enrique Alfonseca and Pilar Rodríguez. 2003. *Generating Extracts with Genetic Algorithms*, Springer Berlin Heidelberg, Berlin, Heidelberg, pages 511–519.

- Regina Barzilay and Michael Elhadad. 1999. Using lexical chains for text summarization. *Advances in automatic text summarization* pages 111–121.
- Aurélien Bossard and Christophe Rodrigues. 2011. Combining a multi-document update summarization system cbseas with a genetic algorithm. In Ioannis Hatzilygeroudis and Jim Prentzas, editors, *Combinations of Intelligent Methods and Applications*, Springer Berlin Heidelberg, volume 8 of *Smart Innovation, Systems and Technologies*, pages 71–87.
- Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR'98: Proceedings of the 21st ACM SIGIR Conference*, pages 335–336.
- H. P. Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM* 16(2):264–285.
- Güneş Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*.
- Pierre-Etienne Genest, Guy Lapalme, and Mehdi Yousfi-Monod. 2009. Hextac: the creation of a manual extractive run. In *Proceedings of the Second Text Analysis Conference*. Gaithersburg, Maryland, USA.
- Dan Gillick and Benoit Favre. 2009b. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*. Association for Computational Linguistics, pages 10–18.
- Dan Gillick, Benoit Favre, Dilek Hakkani-tr, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009a. The ICSI/UTD summarization system at TAC 2009. In *Proceedings of Workshop on Summarization task at TAC 2009 conference*.
- Kai Hong, John Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland.
- Chen Li, Xian Qian, and Yang Liu. 2013. Using supervised bigram-based ilp for extractive summarization. In *ACL 2013*. The Association for Computer Linguistics, pages 1004–1013.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10.
- Marina Litvak, Mark Last, and Menahem Friedman. 2010. A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of the 48th Annual Meeting of ACL*. ACL '10, pages 927–936.
- Dexi Liu, Yanxiang He, Donghong Ji, and Hua Yang. 2006. Genetic algorithm based multi-document summarization. In Qiang Yang and Geoff Webb, editors, *PRICAI 2006: Trends in Artificial Intelligence*, Springer Berlin Heidelberg, volume 4099 of *Lecture Notes in Computer Science*, pages 1140–1144.
- Annie Louis and Ani Nenkova. 2009. Automatically evaluating content selection in summarization without human models. In *Proc. of the 2009 EMNLP Conference : Volume 1*. ACL, pages 306–314.
- H.P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal* 2(2):159–165.
- David J.C. MacKay and Linda C. Bauman Peto. 1994. A hierarchical dirichlet language model. *Natural Language Engineering* 1:1–19.
- Ryan McDonald. 2007. *A study of global inference algorithms in multi-document summarization*. Springer.
- Kumaresh Nandhini and Sadhu Ramakrishnan Bala-sundaram. 2013. Use of genetic algorithm for cohesive summary extraction to assist reading difficulties. *Applied Computational Intelligence and Soft Computing* 2013:8.
- Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.* 4(2).
- Hitoshi Nishikawa, Kazuho Arita, Katsumi Tanaka, Tsutomu Hirao, Toshiro Makino, and Yoshihiro Matsuo. 2014. [Learning to generate coherent summary with discriminative hidden semi-markov model](#). In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1648–1659. <http://aclweb.org/anthology/C/C14/C14-1156.pdf>.
- Maxime Peyrard and Judith Eckle-Kohler. 2016. Optimizing an approximation of rouge - a problem-reduction approach to extractive multi-document summarization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. Association for Computational Linguistics, volume Volume 1: Long Papers, pages 1825–1836.
- Dragomir R. Radev. 2000. A common theory of information fusion from multiple text sources step one: cross-document structure. In *Proceedings of the 1st SIGdial workshop*. Association for Computational Linguistics, pages 74–83.
- Horacio Saggion, Juan-Manuel Torres-Moreno, Iria da Cunha, and Eric SanJuan. 2010. Multilingual summarization evaluation without human models. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '10, pages 1059–1067.

- Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK.
- Haruka Shigematsu and Ichiro Kobayashi. 2014. Topic-based multi-document summarization using differential evolution for combinatorial optimization of sentences .
- Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. 2012. [Large-margin learning of sub-modular summarization models](#). In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, EACL '12, pages 224–233. <http://dl.acm.org/citation.cfm?id=2380816.2380846>.
- Hiroya Takamura and Manabu Okumura. 2010. Learning to generate summary as structured output. In Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, editors, *CIKM*. ACM, pages 1437–1440.
- Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.* 22(2):179–214.