

A Contextual Query Expansion Approach by Term Clustering for Robust Text Summarization

Massih R. Amini and Nicolas Usunier

Laboratoire d'Informatique de Paris 6

104, Avenue du President Kennedy

75016 Paris, France

{amini,usunier}@poleia.lip6.fr

Abstract

This paper describes the different steps which lead to the construction of the LIP6 extractive summarizer. The basic idea behind this system is to expand question and title keywords of each topic with their respective cluster terms. Term clusters are found by unsupervised learning using a classification variant of the well-known EM algorithm. Each sentence is then characterized by 4 features, each of which uses bag-of-words similarities between expanded topic title or questions and the current sentence. A final score of the sentences is found by manually tuning the weights of a linear combination of these features ; these weights are chosen in order to maximize the Rouge-2 AvF measure on the Duc 2006 corpus.

1 Introduction

In this paper, we describe the LIP6 extractive summarizer. Our approach is built upon a combination of an unsupervised learning strategy for title and question topic expansion and some handcrafted sentence scoring heuristics. These heuristics are based on IR-like similarity measures without any NLP knowledge. The creation of the summary follows then four steps:

1. For each topic, we used the Marcu's alignment technique (Marcu, 1999) between document sentences and the title of the topic to produce a small set of candidate sentences. In this

first step, about $\frac{3}{4}$ of the document sentences are discarded.

2. The remaining sentences are then scored using a linear combination of 4 heuristic features.
3. A first summary is then produced by selecting the 10 highest scored sentences with respect to this combination.
4. A post-processing step is then applied to discard redundant sentences and to get the final summary with at most 250 words.

The key point of our approach is in step 2 and it relies on

- An unsupervised learning technique to generate topic-specific clusters of words. These clusters allow us to augment the initial bag of words representation of both the topic title and the question keywords. The most important features use similarity measures between this new representation of the topic and the document sentences.
- A manual tuning of the weights of the linear combination is based on the maximization of the Rouge-2 AvF measure upon the Duc 2006 corpus.

In the remainder of the paper, we fully describe all the steps of our system, so that it can easily be re-implemented and serve as a baseline for future research on summarization. Section 2 describes the preprocessing applied to the topics and the corresponding documents. Section 3 describes the algorithm used to obtain the clusters of words for each

topic. In section 4 we present a variant of Marcu’s alignment technique that we have used to create candidate sentences for summaries. The features used to rank these final sentences are presented in section 5. Finally, in section 8 we discuss the results we obtained on both Duc 2006 and Duc 2007 collections.

2 Preprocessings

For each topic and their associated documents, we independently applied a sequence of different preprocessing steps. These preprocessings allow to define the elementary units associated with each topic and its related summary:

- *sentences*: are the elementary text-span units when creating a summary. They are obtained by applying the NIST’s sentence segmenter¹ to all documents, before any other preprocessing.
- *documents*: although each topic comes with a set of relevant documents, some contain just 1 sentence and are too small to be indicative in the subsequent processing steps of our system. For each topic, we then define a new set of documents, by merging documents with less than 4 sentences.

From now on, any reference to the documents associated with a topic corresponds to this new set of documents. These new documents will be the elementary unit in the first steps of the processing of each topic: first they will be used to characterise each term before applying the word-clustering algorithm (section 3), and, secondly in each document, candidate sentences for the summary will be identified using Marcu’s alignment technique (section 4).

- *tokens* and *vocabulary*: for each topic, a different vocabulary is created. The vocabulary is the set of tokens appearing in either the current topic (title or questions) or the associated documents. Tokenization is applied to both the topic (title and questions) and its corresponding documents after transforming all characters to thier lowercase. The tokens we consider are all sequences of alphanumeric characters which (1) appear at least in three different documents,

(2) do not belong to the stop-list of the CACM collection² (provided by the University of Glasgow), and (3) contain at least one non-numeric character.

The tokens are the elementary semantic units. From now on, we will use interchangeably the words *token*, *word* or *term*. With this tokenization, the mean number of tokens per document in Duc 2006 collection is 316.60.

3 Term Clustering

For each topic, we cluster vocabulary terms based on their co-occurrence in the documents associated to the current topic with an algorithm similar to the one described in (Caillet et al, 2004). The underlying assumption of this approach is that words which tend to appear in the same documents with the same frequency are semantically or topically related. These clusters will then be used in section 5 to expand topic title or question keywords and they constitute the core of our summarization approach.

The remainder of this section describes the algorithm we have used to obtain, for each topic, the term clusters, and shows some term clusters that we obtained for few topics.

3.1 Notations

For the current topic, we denote by $V = \{w_j\}_{j \in \{1, \dots, |V|\}}$ the set of its $|V|$ vocabulary terms, $D = \{d_i\}_{i \in \{1, \dots, n\}}$ the set of n documents containing the answer to the topic question. We further denote by C the current partition of terms found by the algorithm.

3.2 The Algorithm

Term clustering is based on an unsupervised learning technique to discover groups of words which tend to have similar occurrence statistics in documents associated to the current topic. The learning technique is based on a Classification-Expectation-Maximization (CEM) algorithm (Celeux et al, 1982) which is an extension of the well-known EM algorithm (Dempster et al, 1977) in which a Classification step is performed between each Expectation and Maximization steps

¹<http://duc.nist.gov/duc2004/software/duc2003.breakSent.tar.gz>

²http://ir.dcs.gla.ac.uk/resources/test_collections/cacm/

(Algorithm 1). Terms belonging to the same cluster will have similar representation in the document space, which means that they frequently co-occur simultaneously in these contexts. The algorithm is applied independently to each document collection of the current topic. For simplicity in notations, we assume here that a topic has been chosen and is kept fixed.

The algorithm is based on a representation of each vocabulary word w as a bag-of-documents:

$$\vec{w} = \langle tf(w, d_i) \rangle_{i \in \{1, \dots, n\}}$$

where \vec{w} is the representative vector of the word w , n is the number of documents associated to the current topic, d_i is the i -th document, and $tf(w, d_i)$ is the term frequency of word w in document d_i .

By fixing the number of term clusters K , the algorithm associates then every word w in the vocabulary to one (and only one) of the K clusters $1, \dots, K$, based **(1)** on a probabilistic generative model of \vec{w} , and **(2)** a simplifying assumption that if w and w' are two different words, \vec{w} and \vec{w}' are generated independently from one another by the generative model.

More formally, we assume first that each term w is generated by a mixture density

$$p(\vec{w} | \Theta) = \sum_{k=1}^K \pi_k p(\vec{w} | c = k, \theta_k) \quad (1)$$

where, for each k , θ_k is a set of parameters (to be determined by the algorithm) which characterizes the cluster k , Θ is the set of all the model parameters, $p(\vec{w} | c = k, \theta_k)$ is the probability of generating w knowing that it belongs to the cluster k , and $\pi_k = p(c = k | \Theta)$, the probability that a randomly generated word belongs to cluster k .

The second assumption is that each term belongs to one and only one term cluster. Formally, we associate an indicator vector class $t_i = \{t_{hi}\}_h$ to each term $w_i \in V$ such that :

$$\forall w_i \in V, \forall k, y_i = k \Leftrightarrow t_{ki} = 1 \text{ and } \forall h \neq k, t_{hi} = 0$$

3.3 Concept learning

The term clusters C are found by searching for parameters Θ which maximize the *complete* data log-likelihood:

$$\mathcal{L}_{CML}(C, \Theta) = \sum_{w_j \in V} \sum_{k=1}^K t_{kj} \log p(\vec{w}_j, y = k, \Theta)$$

Here, the class indicator vectors t are model pa-

Algorithm 1: CEM algorithm

Input :

- An initial partition $C^{(0)}$ is chosen at random and the class conditional probabilities $p(w | y = k, \theta_k^{(0)})$ are estimated on the corresponding classes.
- $l \leftarrow 0$

repeat

- **E-step**: Estimate the posterior class probability that each term w_j belongs to $C_k^{(l)}$:

$$\forall w_j \in V, \forall k \in \{1, \dots, K\},$$

$$\mathbb{E}[t_{kj}^{(l)} | \vec{w}_j; C^{(l)}, \Theta^{(l)}] = \frac{\pi_k^{(l)} p(\vec{w}_j | y=k)}{p(\vec{w}, \Theta^{(l)})}$$

- **C-step**: Assign each $w_j \in V$ to the cluster $C_k^{(l+1)}$ with maximal posterior probability according to $E[t | w]$. Let $C^{(l+1)}$ be the new partition.

- **M-step**: Estimate the new parameters $\Theta^{(l+1)}$ which maximize

$$\mathcal{L}_{CML}(C^{(l+1)}, \Theta^{(l)})$$

- $l \leftarrow l + 1$

until convergence of \mathcal{L}_{CML} ;

Output : Term clusters C

rameters and are estimated together with the Θ . In our experiments, we supposed that terms are generated independently from the mixture density (1) where each mixture component $p(\vec{w} | y)$ obeys a naive Bayes model. The parameters Θ of this model are the set of class prioris $\pi_k = p(y = k)$ and the probability of documents d_i with respect to different clusters $\{p_{ik}\}_{i \in \{1, \dots, n\}, k \in \{1, \dots, K\}}$. Under these assumptions, $p(\vec{w} | y = k) = \prod_{i=1}^n p_{ik}^{tf(w, d_i)}$.

Differentiating \mathcal{L}_{CML} in turn with respect to π_k and p_{ik} and using Lagrange multipliers to enforce the constraints $\sum_k \pi_k = 1$ and $\forall k, \sum_{i=1}^n p_{ik} = 1$,

we get the maximum likelihood estimates of π_k and p_{ik} :

$$\pi_k = \frac{\sum_{j=1}^{|V|} t_{kj}}{|V|}$$

$$p_{ik} = \frac{\sum_{j=1}^{|V|} t_{kj} \times tf(w_j, d_i)}{\sum_{j=1}^{|V|} \sum_{i=1}^n t_{kj} \times tf(w_j, d_i)}$$

3.4 Setting and examples of term clusters

In our experiments, we have set for each topic the number of term clusters to one fifteenth its vocabulary size. This setting is mostly experimental and this number seemed to produce accurate clusters in general.

Table 1 shows some clusters found for two topics in Duc 2006 and 2007 data sets. An interesting point with these term clusters is that they disambiguate some name entities based on whether they co-occur with specific terms related to the topic. For example, for the topic D0705 *Herri Batasuna* has been found to co-occur frequently in the same documents as the terms *ETA*, *Basque*, *Separatist*. More generally, a manual observation of the clusters tend to show that the motivating assumption behind term clustering (i.e. words which appear in the same document with the same frequencies are semantically or topically related) is valid.

4 Marcu alignment technique to remove non-informative sentences

When all the previous processings are done, the first step of our summarizer consists in removing a large majority of non-informative sentences of each topic. The candidate sentences for final summaries are obtained here using the Marcu’s alignment technique (Marcu, 1999), where we align each document with the questions associated to the current topic. This algorithm results in extracting, for each document, the subset of its sentences which is the most *similar* to the question with the underlying assumption that, in each document, the smallest set of sentences which contains the answer to the current question is also the one which has the maximal semantic similarity.

After this step, all the summaries obtained for a topic are concatenated. This new set of sentences associated to a given topic will be used in the remaining processings.

In the remainder of this section, we describe the algorithm used and the experimental settings, as well as some results showing that this first step discards a vast majority of the sentences while keeping the informative ones.

4.1 Algorithm

More precisely, following (Marcu, 1999), the similarity between a set of sentences S and the questions associated with a topic is defined using a bag-of-words representation of them, as follows:

$$Sim(S, Q) = \frac{\sum_{w \in S \cap Q} c(w, S)c(w, Q)}{\sum_{w \in S} c^2(w, S) \sum_{w \in Q} c^2(w, Q)}$$

where $w \in S$ (resp. $w \in Q$) denotes the presence of term w in the set of sentences S (resp. in the question Q), and $c(w, S)$ (resp. $c(w, Q)$) is the term weight associated to w in the set of sentences S (resp. in the questions Q). The term weighting scheme we have chosen is the following:

$$c(w, Z) = tf(w, Z) \times \log(df(w))$$

where $tf(w, Z)$ is the term frequency of w in Z (with $Z = S$ or $Z = Q$) and $df(w)$ is the number of documents (associated to the current topic) in which w appears.

Marcu’s algorithm (Marcu, 1999) is then applied to find the set of candidate sentences to be included in the summary of a topic. This algorithm is an iterative one which initially sets S to be the set of all sentences in a document. At each iteration it then removes a sentence from the current set such that its removal maximizes the similarity between Q and the rest of sentences in that set. The algorithm stops when the removal of any sentence implies a decrease of similarity between Q and the remaining set. The behavior of the algorithm is plotted in figure 1.

4.2 Performance of Marcu’s algorithm

Table 2 shows the Rouge-1 and Rouge-2 average scores of summaries made by concatenating all sentences in all documents in each topic before and after the Marcu’s alignment technique for Duc 2006 task. We can see that the average F-measures of the overall sentences after aligning have notably increased for both Rouge-1 and Rouge-2 measures without a

D0614 - Quebec independence
Cluster containig <i>quebec</i>: majority minister future prime chretien canadians national federalist believe stay face poll confederation unity center legislature uncertainty canada <u>quebec</u> province
Cluster containing <i>independence</i>: separatists united clear <u>independence</u> leaders need opinion states public votes despite lucien create negotiations officials bouchard independent opposition france
D0705 - Basque separatism
Cluster containig <i>basque</i> and <i>separatism</i>: <u>basque</u> people separatist armed region spain <u>separatism</u> eta independence police france batasuna nationalists herri bilbao killed

Table 1: Two term clusters found with CEM in Duc 2006 and Duc 2007.

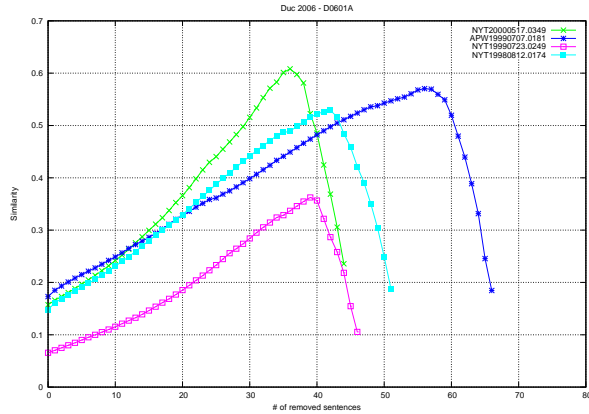


Figure 1: Evolution of the similarity measure with respect to the number of sentences removed using (Marcu, 1999) alignment algorithm for some documents in D0601.

Average # of	Before align ^t	After align ^t
sent. per topic	655.44	156.7
words per sent.	12.78	9.78
Rouge-1 Av_R	0.96767	0.89553
Rouge-1 Av_P	0.01877	0.06129
Rouge-1 Av_F	0.03664	0.11473
Rouge-2 Av_R	0.56947	0.44690
Rouge-2 Av_P	0.01073	0.05299
Rouge-2 Av_F	0.02096	0.09474

Table 2: Rouge-1 and Rouge-2 average scores of the sets of sentences present in all documents in each topic before and after Marcu’s alignment technique for Duc 2006 task.

big loss in average recall. Furthermore the alignment technique does not change the sentence-word distribution (figure 2). For Duc 2006 task, in both cases the sentence length in filtered words is narrowly distributed around 10 – 12 words.

More generally, we have tried various variations of this step: alignment with the topic title instead of the topic questions (or alignment with both of them), as well as other similarity measures like *tf.idf*. The settings we have presented in this section obtained the best performance on the DUC 2006 corpus.

5 Sentence features

Our sentence feature generation is based on the mapping between each sentence in the final sentence pool and the queries obtained from each question and topic title. We considered three queries namely:

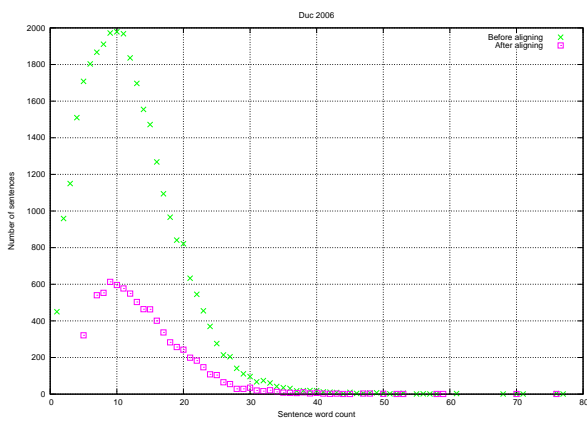


Figure 2: Sentence - filtered word distribution

q_1 obtained from question keywords, q_2 and q_3 obtained by extending respectively question keywords and topic title with terms in their respective term-

clusters. Each ranking feature is then defined as:

$$f : \{queries\} \times \{sentences\} \mapsto \mathbb{R}$$

$$f(q, s) = score(q, s)$$

We have tested different scoring schemes and found the following features as the most performing ones:

Feature	Query	Score
F_1	q_1	$common_terms(q_1, s)$
F_2	q_1	$cosine(q_1, s)$
F_3	q_2	$ldf(q_2, s)$
F_4	q_3	$ldf(q_3, s)$

Where $common_terms(q, s)$ is the number of common terms within a query and a sentence, $cosine(q, s) = \frac{\sum_{w \in q \cap s} c(w)_q c(w)_s}{\sum_{w \in q} c(w)_q^2 \sum_{w \in s} c(w)_s^2}$ where $c(w)$ is the same term weight than the one considered in the Marcu’s alignment algorithm and $ldf(q, s) = \sum_{w \in q \cap s} \log(df(w))$. Where $df(w)$ is the document term frequency estimated before merging documents in the previous step.

Table 3 shows the Spearman rank order correlation coefficient between these four features. This correlation coefficient is computed as follows:

$$r_s = 1 - \frac{6 \sum_i d_i^2}{n(n^2 - 1)}$$

Where d is the difference in rank between two ordered lists and n the number of sentences in each sentence pool for each topic. It is to be noted that in the Spearman correlation only the order of the data is important, not the level, therefore extreme variations in expression values have less control over the correlation. The low correlation values suggest that there are low linear relationship between different ordered lists obtained from these features. By combining them one might then expect to find more relevant sentences in the top of the ordered list obtained from the combination than each individual ordered list related to each feature.

6 Feature Combination

Considering the low correlation of the sentence features and the potential gain of combining them, we have employed different strategies to find weights to create a performing linear combination of them.

Features	F_1	F_2	F_3	F_4
F_1	*	0.198	0.186	0.141
F_2	*	*	0.095	0.086
F_3	*	*	*	0.123

Table 3: Spearman correlation between different ordered lists obtained from each feature.

We first tried to construct a training set to learn these weights but this strategy did not worked well. We then tried to merge each ordered list using the weighted Borda fuse algorithm (Aslam et al, 2001) which did not give much more success. We then manually found the feature weights by maximizing the Rouge-2 AvF measure upon the DUC 2006 corpus for summaries constituted from the first 10 sentences having highest score with respect to this combination. The best linear combination of the normalized features (obtained by dividing them by their highest value) with respect to this strategy was :

$$\forall s, Score(s) = 0.2 * F_1 + 0.04 * F_2 + 0.4 * F_3 + 0.36 * F_4$$

7 Postprocessings

To reduce redundancy, we followed (Conroy et al, 2006), by gathering the 10 highest score sentences with respect to the previous scoring scheme and having no more than 8 terms in common. The lead sentence in the final summary was the one having the highest score. We then add one by one sentences using the Traveling Salesperson formulation as it was formulated in (Conroy et al, 2006) with the final summary length constraint of 250 words.

8 Discussion

The LIP6 summarizer id was 4. As shown in table 4 our system did well on Duc 2007 with respect to the Rouge-2 and Rouge-SU4 measures. We also notice an increase in performance of our system from Duc 2006 to Duc 2007. We expect that the difference in performance may be because the model summaries were more similar to the true extract summaries for each topic this year.

Furthermore, we believe that combining sentence features is an essential tool to make good summaries. Learning the feature weights in different classification and ranking settings has shown to be

DUC 2006						
Scoring	Rouge-2			Rouge-SU4		
	Av-R	Av-P	Av-F	Av-R	Av-P	Av-F
F_1	0.07428	0.07431	0.07429	0.129124	0.12904	0.12908
F_2	0.07865	0.07894	0.07879	0.12734	0.12728	0.12731
F_3	0.08512	0.08517	0.08514	0.14253	0.14312	0.14282
F_4	0.08856	0.08861	0.08858	0.14521	0.14231	0.14374
$\sum_{i=1}^4 \alpha_i F_i$	0.09114	0.09108	0.09111	0.14932	0.14886	0.14908
DUC 2007						
$\sum_{i=1}^4 \alpha_i F_i$	0.11887	0.11894	0.11886	0.16999	0.17027	0.17007

Table 4: Rouge-2 and Rouge-SU4 results of the LIP6 system obtained in DUC 2006 and Duc 2007

very efficient in the literature. The main difficulty is how to find gold extracts in order to constitute the training set. It may be worth to consider this issue and to make gold extracts manually for some topics. Automatic tools which help to find these gold extracts can also be considered (Amini, 2000; Amini et al, 2003).

Acknowledgements

The authors would like to thank Jean-François Pessiot for his helpful comments. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors view.

References

- Massih R. Amini 2000. *Interactive Learning for text summarization*, Proceedings of the PKDD workshop on Machine Learning and Textual Information Access.
- Massih R. Amini and Patrick Gallinari 2003. *Semi-Supervised Learning with Explicit Misclassification Modeling*, Proceedings of the 18th International Joint Conference on Artificial Intelligence, 555–560.
- Javed A. Aslam and Mark Montague 2001. *Models for metasearch*, SIGIR, 276–284.
- John M. Conroy, Judith D. Schlesinger, Dianne P. O’Leary and Jade Goldstein 2006. *Back to Basics: CLASSY 2006*, Proceedings of DUC’06.
- Marc Caillet, Jean-François Pessiot, Massih-Reza Amini and Patrick Gallinari. 2004. *Unsupervised Learning with Term Clustering for Thematic Segmentation of*

Texts Proceedings of the 7th Recherche d’Information Assistée par Ordinateur (RIAO’04), 648–656.

Gilles Celeux and Gilles Govaert. 1992. *A Classification EM algorithm for clustering and two stochastic versions* Journal of CSDA, 14(3): 315–332

A. Dempster, N. Laird and D. Rubin. 1977. *Maximum likelihood from incomplete data via the EM algorithm*, *Journal of Royal Statistical Society*, 39(1): 1–38.

Daniel Marcu. 1999. *The Automatic Construction of Large-Scale Corpora for Summarization*. Proceedings of the 22nd ACM SIGIR Conference, 137–144.

Appendix

DUC 2007			
Id	Mean	95% low. C.I.	95% upp. C.I.
D	0.17175	0.15322	0.19127
C	0.14993	0.13372	0.16741
J	0.14141	0.12265	0.16274
G	0.13903	0.12312	0.15385
E	0.13764	0.12413	0.15315
B	0.13740	0.11372	0.16061
F	0.13739	0.12097	0.15530
A	0.13430	0.11765	0.15108
I	0.13328	0.11017	0.15481
H	0.12702	0.11448	0.13995
15	0.12285	0.11800	0.12768
4	0.11886	0.11467	0.12351
29	0.11725	0.11245	0.12225
24	0.11605	0.11040	0.12133

Table 5: Average F score of Rouge-2 scores

DUC 2007			
System Id	Mean	95% lower condifence intervals	95% upper condifence intervals
D	0.21461	0.20154	0.22922
C	0.19846	0.18350	0.21478
J	0.19378	0.17834	0.21139
E	0.19266	0.18147	0.20490
F	0.19165	0.17905	0.20506
A	0.18902	0.17749	0.20182
G	0.18761	0.17638	0.19886
B	0.18620	0.16685	0.20543
H	0.18044	0.17067	0.18967
I	0.18016	0.16292	0.19648
15	0.17470	0.16997	0.17939
24	0.17304	0.16800	0.17769
4	0.17007	0.16646	0.17381
29	0.16635	0.16163	0.17113

Table 6: Average F score of Rouge-SU4 scores

DUC 2007			
System Id	Avg. Content	System Id	Avg. Linguistic
D	4.94	G	4.93
G	4.89	E	4.90
I	4.89	I	4.89
F	4.72	F	4.88
C	4.67	D	4.86
E	4.67	A	4.80
H	4.67	J	4.78
A	4.61	C	4.76
B	4.56	H	4.76
J	4.50	B	4.48
4	3.80	23	4.11
23	3.31	4	3.82
14	3.13	14	3.67

Table 7: Linguistic scores