Combining a mixture language model and Naive Bayes for multi-document summarisation

Wessel Kraaij, Martijn Spitters, Martine van der Heijden

TNO-TPD
P.O. Box 155, 2600 AD Delft
The Netherlands

Abstract

The TNO system for multi-document summarisation is based on an extraction approach. We combined two statistical methods for sentence selection with a variant of the MMR algorithm. After sentence segmentation, each sentence is scored on the basis of two probabilistic models. The first model scores sentences based on a (generative) unigram language model, which is a mixture of a cluster model, a document model and a background model, this score is compared to the probability that the sentence is generated by just the background model. The resulting log likelihood ratio is normalised on the basis of sentence length. The second model is a simple Bayesian model based on several non-content sentence features: sentence position, sentence length and cue phrases. The scores of both models yield a likelihood ratio score which can are combined to yield a more reliable salience score. Finally, the summary is constructed by selecting the most salient sentence and add sentences which are both salient and do give new information.

1 Introduction

This paper describes the design and development of a system for multi-document summarisation based on probabilistic methods. The system has been tested on the DUC 2001 test-set. Section 5 presents a preliminary evaluation of the results.

Document summarisation is a new research area at TNO TPD. For DUC 2001 we built a system from scratch based on probabilistic models. We had good experience with the application of probabilistic models for different IR tasks: ad hoc retrieval (including cross language and spoken document retrieval), filtering, topic tracking etc. [5, 4, 8]. Our goal for DUC 2001 was to investigate to what extent these minimalist models can be applied for a summarisation task. Of course, the probabilistic framework has some limitations (bag of words assumption, sentences independent, no discourse structure), but nevertheless we were confident to reach an acceptable baseline within the relatively short time constraints for DUC 2001. At a later stage, we plan to include deeper linguistic analysis steps into the framework. At this point we constrain ourselves to an "extraction" based approach to summarisation, i.e. a summary consists of extracted text segments of the original textual data. A more mature system would contain a real abstraction/condensation component, which requires a deeper level of interpretation.

2 DUC tasks

DUC (Document Understanding Conference) is a new series of evaluation conferences which has the goal to stimulate progress in the field of automatic summarisation. DUC 2001 defined three tasks: i) summarise a single document, ii) summarise a cluster, iii) exploratory summarisation. We chose to concentrate on the second task, and tested our multi-document system on single documents as an extra experiment.

The data provided by NIST consisted of 30 training clusters, each containing between 3 and 20 documents on the same topic, with an average of 10 documents. The documents are all English news articles from the following newspapers: Wall Street Journal, San Jose Mercury News, LA Times, Financial Times, AP Newswire and Foreign Broadcast Information Service. They are at least 10 sentences long, with no maximum. For each document there was a per-document summary with an approximate length of 100 words. And for each cluster, consisting of related documents, four multi-document summaries were provided, with lengths of 400, 200, 100 and 50 words. The test data consisted of another set of 30 clusters of documents.

3 Design of a probabilistic sentence extraction system

Previous research on extraction based approaches has shown the effectiveness of several non-content or *surface* features for the determination of a sentence's salience: position in text, the usage of cue phrases, sentence length etc[3]. We decided to base our sentence extraction algorithm on a combination of both a content based algorithm (a unigram language model, which could be considered as a more principled way to rank sentences on tf.idf [7]) and a classifier based on non-content features like sentence length.

The combined model determines a salience value for each extracted sentence. This ranked list of sentences forms the input for the summary generation module. This module tries to generate a summary which consists of the most salient sentences, with minimal redundancy and maximal coherence/readability. Since we do not have a deeper meaning representation of the extracted sentences, we can only use very shallow techniques to meet the latter constraints. The maximum marginal relevance (MMR) criterion [2] was adapted for our system in order to minimise redundancy of the produced summaries.

3.1 Unigram language model for content based salience

The salience of a sentence $S = T_1, T_2, ..., T_n$ w.r.t. a document D_k (the within document salience) can be modelled as the probability that the sentence is generated by a unigram model corresponding to that document, assuming independence between the individual terms.

$$P(T_1, T_2, ..., T_n | D_k) = \prod_{i=1}^{n} P(T_i | D_k) \tag{1}$$

The salience of a sentence with respect to a cluster of documents (the within cluster salience) can be modelled along the same lines (i.e. replace document D_k by cluster C_j in formula (1)).

Our hypothesis about multi-document summarisation is that a "good" sentence is both salient for a document and for the corresponding cluster. $P(S|D_k)$ and $P(S|C_j)$ are combined in a straightforward way: by linear interpolation. The resulting mixture was interpolated with a third component: the background model $P(S|GE)$ in order to smooth probability estimates[1]. This results in the following mixture model:

$$P(S|D_k, C_j) = \prod_{i=1}^{n} (\lambda P(T_i|D_k) + \mu P(T_i|C_j) + (1 - \mu - \lambda) P(T_i)) \tag{2}$$

As a final step we applied two normalisation steps in order to be able to use the probability as a metric across sentences and applied a logarithm in order to convert the product into a summation.

$$LLR_{\text{content}}(S) = LLR(S|D_k, C_j) = \sum_{i=1}^{n} \log\left(\frac{\lambda P(T_i|D_k) + \mu P(T_i|C_j) + (1 - \mu - \lambda) P(T_i)}{P(T_i)}\right)^{-n} \tag{3}$$

Formula (3)shows the final model which can be paraphrased as the (geometric) mean of the log likelihood ratio of a sentence given the mixture model and given the background model.

The conditional probabilities were estimated using maximum likelihood procedures. The documents, clusters and corpus of general English were stemmed (Porter) and stopped in order to reduce morphological variation and eliminate non-content words.

[1]We used the TREC8 Ad-Hoc document collection for parameter estimation of the background model

3.2 Naive Bayes classifier for surface based salience

Since surface features are important predictors of salience, we decided to integrate them into our system. We decided to work with a Naive Bayes classifier, because Kupiec [6] had reported good results and NB classifiers are extremely easy to implement.

Five clusters of the DUC training data (splitted into 4132 sentences) were annotated in order to get an idea about the relative importance of different feature types for the prediction of salience in the news domain. As a first step, documents were segmented into sentences, using a rule and abbreviation lexicon based sentence splitter developed at RALI, Université de Montréal. Subsequently, each of the sentences was scored on salience using a scale of five values, varying from 'completely irrelevant (-2)' to 'highly relevant (2)'. A sentence was considered relevant if it appeared literally or almost literally in the provided example summaries, or if it contained much information relevant to the topic of the cluster. The annotator read all documents of a cluster, and the 50,100,200 and 400 word cluster summary. These summaries were taken as the “gold standard”.

After this salience annotation, we investigated which features were good in predicting salience or non-salience. Salience values +1 and +2 were grouped in the class *salient*, salience values 0, -1 and -2 were grouped in the class *non salient*. Three features were selected: *cue phrase*, *sentence length* and *first sentence*.

first sentence (fs) The best predictive feature is whether or not a sentence is the first sentence in a document, because in news articles the first sentence often describes briefly what the article deals with. After annotating five clusters of training documents it became clear that a lot of first sentences contain important information.

cue phrase (cp) Phrases like ‘conclusion’ or 'in particular' are often followed by important information. Thus, sentences that contain one or more of these so-called *cue phrases* are considered more relevant than sentences without cue phrases.

sentence length (sl) Another important feature is the *sentence length*. Very short sentences like 'Marshall joined it.' are often not very informative. Very long sentences are often more relevant than medium length sentences.

After feature selection, the same set of sentences was annotated for the selected features. The cue phrase feature and first sentence feature are binary: a sentence either does or does not contain a cue phrase, and either is or is not the first sentence in a document. We decided to make the sentence length feature tripartite since it is impossible to draw an absolute boundary between short and non-informative on the one hand, and long and informative on the other. The sentence length is calculated after filtering out non-content-words. After stopping a sentence containing 0 to 6 words is considered short (0), a medium sentence has a length between 6 and 14 (1) and a sentence length higher than 14 non-stop words is considered long (2). These boundaries were empirically chosen (cf. Section 4).

In the naive Bayes approach, the probability that a sentence will be included in the summary given a feature vector can be calculated using Bayes' rule [6].

$$P(s \in S|\bar{\mathbf{x}}) = \frac{P(\bar{\mathbf{x}}|s \in S)P(s \in S)}{P(\bar{\mathbf{x}})} \tag{4}$$

In formula (4), $\bar{\mathbf{x}}$ is a shorthand for $P(\bar{\mathbf{X}} = \bar{\mathbf{x}}$ i.e. $\bar{\mathbf{X}}$ is a random variable with as value a vector of features $\bar{\mathbf{x}} = (x_1, x_2, ..., X_k)$. Note that $s \in S$ refers here to the probability that a sentence s is part of a summary S (Assuming statistical independence of the features):

$$P(s \in S|\bar{\mathbf{x}}) = \frac{\prod_{j=1}^{k} P(x_j|s \in S)P(s \in S)}{\prod_{j=1}^{k} P(x_j)} \tag{5}$$

“Training” the naive Bayes classifier consisted of estimating the conditional probabilities $P(x_j|s \in S)$ and the marginal probabilities $P(x_j)$ on the 4132 annotated sentences.

Table 1 shows the predictive power of the different feature values. The feature “first sentence” has a very strong predictive power. Cue phrases exhibit only a small correlation with salient sentences, even smaller than long sentences. The sentence length feature is especially effective to predict non-salience. This is probably due to the relative high frequency of one, two word “sentences” in the

x_j	$P(x_j \| s \in S)/P(x_j)$
cp=1	1.34680
cp=0	0.91670
fs=1	6.36960
fs=0	0.92350
sl=0	0.00001
sl=1	0.73649
sl=2	1.85010

Table 1: Predictive power of the features

annotated corpus. Sometimes these are real sentences, often these short sentences are due to a sentence segmentation problem, e.g. a list of senate members is formatted with semicolons. The large variety of interpunction in the test collection (e.g. broadcast transcripts) made it impossible to reach a perfect accuracy using generic splitting rules. We used the figures in table 1 for the generation of all our summary sizes, i.e. the classifier was trained independent of the desired compression rate, which is probably not optimal.

3.3 Combining the models

Combination of the models described in the previous sections in order to compute a single salience value is not trivial. The salience score resulting from the mixture language model is an average likelihood ratio based on two language models, while the Naive Bayes classifier computes the posterior probability that a sentence will be included in the summary. The former function is on a ratio scale while the latter is a probability ranging between 0 and 1. Since applying logistic regression on the likelihood ratio is difficult since there is hardly training data, we propose an ad-hoc solution to map both scores to approximately the same domain. We propose to use the posterior log-odds of the Naive Bayes classifier $\log O_{\text{surface}} = \log(P(s \in S|\bar{\mathbf{x}})/(1 - P(s \in S|\bar{\mathbf{x}})))$ and interpolate this value with the likelihood ratio:

$$R_{\text{combi}} = \alpha LO_{\text{surface}}(S_i) + (1 - \alpha) LLR_{\text{content}}(S_i) \qquad (6)$$

3.4 Diversity-based re-ranking of salient sentences

In extraction-based document cluster summarisation the summary must pool sentences from different documents. However, because the documents share the same topic or event they might contain overlapping information. For instance, news stories discussing related events often contain repetition of background information. Selecting sentences merely on their salience with respect to the cluster might result in a summary containing redundant information. Reducing redundancy is therefore an indispensable step in the multi document summarisation process.

In order to provide for this step, we implemented a simple algorithm for diversity-based sentence re-ranking. The algorithm is an adapted version of the Maximal Marginal Relevance (MMR) method by Carbonell and Goldstein [2]. MMR has proven to be an effective method for redundancy reduction in document retrieval and passage selection for summarisation. The MMR criterion selects documents or passages that are both relevant to the query and contain minimal repetition of information already presented to the user.

Our version of MMR works as follows. First, sentences are ranked based on their content-based salience (see 3.1). The sentence with the highest salience becomes the first sentence of the summary. The remaining candidate sentences are re-ranked based on the combination of maximal content based and surface based salience and minimal similarity to the sentence that is already in the summary. The sentence with the highest combined score is added to the summary and the remaining sentences are re-ranked again, because the summary contains new information. This procedure continues until the summary has the desired length. Equation (7) shows our implementation of the MMR criterion.

$$MMR = \arg\max_{S_j \in R \setminus A} [R_{combi}(S_i) - \beta \max_{S_j \in A} LLR(S_i|S_j)] \quad (7)$$

Where S is a sentence, D_k is the document model, C_l is the cluster model, A is the abstract and R is the ranked list of sentences to which the algorithm is applied. The parameter β determines to what extent novelty contributes to the final score.

When the summary has reached the desired length, the algorithm stops. Finally, to improve the readability of the summary, the selected sentences are chronologically ordered.

3.5 Single document summary

Because the model for sentence selection is composed on a mixture of clearly identifiable components, we hypothesised that a simplified model could be effective for single document summaries. We adapted our model for single document summaries by simply removing the cluster component from the mixture model (2). The content-based likelihood ratio conditioned on the document is formalised on formula (8).

$$\mathrm{LR}_{\mathrm{content}} = \mathrm{LR}(S|D_k) = \prod_{i=1}^{n}\left(\frac{\lambda P(T_i|D_k) + (1-\lambda)P(T_i)}{P(T_i)}\right)^{-n} \quad (8)$$

For an optimal performance, the Naive Bayes classifier should have been retrained on a new corpus, annotated with the single-document summaries as the gold standard. We guessed that the classifier trained on the corpus annotated for the multi-document summaries would still be quite effective.

3.6 Related work

Our work follows the approach of Edmundson [3], who recognised that several classes of features (Cue word, Key word, Location and Title) can be effective for the prediction of sentence salience. Edmundson combined the partial scores of the distinct feature classes using a linear combination, where the different factors were hand-tuned. We made a somewhat different distinction: content features i.e. features which predict the salience w.r.t. the topic of the document, and surface features i.e. features that predict salience according to general or domain specific principles. We applied different types of probabilistic models for these classes.

Language models have been applied for an abstracting approach to summarisation by Berger and Mittal [1]. This approach is more advanced, employing separate probabilistic component models for content abstraction and surface realisation. Our approach and the Ocelot system share the basic point of departure of using generative language models. Our system computes the salience of sentences extracted from a document, whereas Ocelot generates the most probable gist given a generative language model given the source document and a trigram language model to ensure readability.

4 Parameter tuning

During the testing phase we discovered that the Bayesian classifier and the generative model play a complementary role. Using only the generative content-based model to rank sentences results in the selection of short sentences with one or two relatively important nouns. On the other hand, the Bayesian classifier gives preferential treatment to sentences that come first in a document, because that feature has very strong predictive power (3.2). When combined in an appropriate manner, the two models cancel out each other's disadvantages.

We performed several runs with different values for the parameters λ, μ, α and β. Only the latter two parameters (see equations 6 and 7) strongly influenced the quality of the summary. Our system appeared to be relatively insensitive for different values of λ and μ (see equations 2 and 3).

5 Results

For a preliminary analysis, we derived some measures from the result files distributed by NIST. The measures are listed in table 2

opg	overall peer grammaticality
opc	overall peer cohesion
opo	overall peer organisation
upg	good unmarked peer units
upm	related unmarked peer units
upb	unrelated unmarked peer units
ur	number of peer units / number of model units
pp	pseudo precision: fraction of good peer units (0-1)
pr	pseudo recall: fraction of covered model units (0-1)
pc	pseudo coverage: mean extent of coverage per model unit (0-4)

Table 2: Description of the measures

Tables 3-12 show the results for each measure for 4 different summary lengths. Each row shows: the mean score of the manually constructed summaries, the scores of the two baseline runs, the average score of all submitted runs and the score of our system.

We will discuss the results of these measures briefly. First for the multi-document summaries:

opg Overall grammaticality does not seem to be a problem, except for the baseline systems. Their short summaries are probably cut off in the middle of a sentence

opc Cohesion is of course stable for baseline 1, since that baseline takes the top part of the last document. For the other systems, cohesion degrades with summary length.

opo Overall organisation is again good for baseline 1. Most systems perform better than baseline 2 (first sentence of the individual douments). Our system performs at an average level, except for the very short summary, where we perform well above average. This is probably due to the fact that our 50 word summaries often consist of just one long sentence.

upg,upm,upb Most unmarked peer units (for all summaries) are of the category “related”.

ur Our system has a low ratio of peer units per model unit. This can probably be explained by the system’s preference for long sentences.

pp Our system scores close to the manual summaries in this category. Especially for longer summaries, the average proportion of peer units that is marked is almost equivalent to a manual summary.

pr The pseudo recall of automatic systems is usually far below the score for manual systems. Baseline 1 (the system that takes the top N of the last documents) performs worse than average for longer summaries, because it does not include material from other documents of the cluster.Our system scores better than average for the 200 and 400 word summaries, but lower than baseline 2 for the short summaries. This could be explained by the fact that the Bayesian classifier is trained for 400 word summaries.

pc The pseudo coverage is the average extent to which model units are covered on a scale of 0-4. The figures for the manual runs show that there is quite some disparity between the model summaries and the manual summaries. The average automatic system scores are comparable to baseline 2, which is a bit a discouraging result. Our system scores lower than baseline 2 for the 50-100 length summaries and better than the baseline for the 100-200 word length summaries.

Our conclusion is that our system has a tendency for longer sentences which hurts performance for short summaries. On the other hand, for the longer summaries, our system performs better than average. One explanation could be the fact that the Bayesian classifier is trained for 400 words summaries. Also a sentence extraction approach is not very well suited for short summaries.

The results for the single document summaries (Table 13) are as follows:

opg Our system scores above baseline (b1: take first hundred words) and system average.

opc Our system scores below the system average. The baseline scores best.

opo Our system scores below the system average. The baseline scores best.

upg,upm,upb Stable picture over all summaries: most unmarked peer units are “related”.

ur Again, our system has a bias for longer sentences.

pp Pseudo precision is quite stable and comparable overall.

pr Pseudo recall is comparable for automatic systems, but considerably lower than the manual summaries.

pc Our system scores lower than the baseline and the system average on the average extent of coverage of model units. This might be related to the bias for long sentences.

Our single-document summarisation system (a simplified version of our system for multi-document summarisation) had not been tuned to single-document summaries. This is probably the reason that our scores are a bit below the baseline on the coverage measures (opo and pc).

	avman	b1	b2	avsys	tno
M-050	3.82	2.86 (-0.96)	2.76 (-1.06)	3.45 (-0.37)	3.54 (-0.28)
M-100	3.78	3.17 (-0.61)	3.36 (-0.42)	3.47 (-0.31)	3.71 (-0.07)
M-200	3.75	3.41 (-0.34)	3.43 (-0.32)	3.48 (-0.27)	3.72 (-0.03)
M-400	3.60	3.28 (-0.32)	3.55 (-0.05)	3.36 (-0.24)	3.70 (+0.10)

Table 3: opg

	avman	b1	b2	avsys	tno
M-050	2.86	2.52 (-0.34)	1.66 (-1.20)	2.10 (-0.76)	2.21 (-0.65)
M-100	2.64	2.52 (-0.12)	1.71 (-0.93)	1.86 (-0.78)	1.79 (-0.85)
M-200	2.81	2.83 (+0.02)	1.86 (-0.95)	1.87 (-0.94)	1.93 (-0.88)
M-400	2.65	2.66 (+0.01)	1.66 (-0.99)	1.79 (-0.86)	1.81 (-0.84)

Table 4: opc

	avman	b1	b2	avsys	tno
M-050	3.35	2.31 (-1.04)	1.93 (-1.42)	2.39 (-0.96)	2.79 (-0.56)
M-100	3.14	2.93 (-0.21)	1.64 (-1.50)	2.02 (-1.12)	1.89 (-1.25)
M-200	3.11	3.10 (-0.01)	1.50 (-1.61)	1.85 (-1.26)	1.76 (-1.35)
M-400	3.14	2.86 (-0.28)	1.52 (-1.62)	1.80 (-1.34)	1.78 (-1.36)

Table 5: opo

6 Conclusions

We have succeeded in building a baseline system which performs quite reasonable for the 200 and 400 word multi document summarisation task. A sentence extraction approach is not optimal for the 50-100 word multi document summaries. The short summary task really requires some text compaction/aggregation component.

We think that combining a unigram LM based approach (capturing content) with a Bayesian classifier based on ”surface features” is a new approach to document summarisation. The Bayesian classifier helps to compensate for some undesired properties of the LM based approach, while the LM based salience score helps to select sentences beyond the first sentence of a document. However, our system needs more task specific tuning. A global parameter setting for all the tasks proved not optimal for the single document task.

	avman	b1	b2	avsys	tno
M-050	0.40	0.03 (-0.37)	0.03 (-0.37)	0.14 (-0.26)	0.00 (-0.40)
M-100	0.60	0.24 (-0.36)	0.07 (-0.53)	0.21 (-0.39)	0.04 (-0.56)
M-200	0.63	0.38 (-0.25)	0.21 (-0.42)	0.30 (-0.33)	0.17 (-0.46)
M-400	0.77	0.21 (-0.56)	0.10 (-0.67)	0.39 (-0.38)	0.44 (-0.33)

Table 6: upg

	avman	b1	b2	avsys	tno
M-050	2.61	3.10 (+0.49)	3.17 (+0.56)	2.71 (+0.10)	2.14 (-0.47)
M-100	3.00	3.21 (+0.21)	2.93 (-0.07)	3.03 (+0.03)	2.82 (-0.18)
M-200	3.19	2.97 (-0.22)	3.25 (+0.06)	3.19 (+0.00)	3.38 (+0.19)
M-400	3.19	3.48 (+0.29)	3.24 (+0.05)	3.26 (+0.07)	3.15 (-0.04)

Table 7: upm

	avman	b1	b2	avsys	tno
M-050	0.23	0.72 (+0.49)	0.72 (+0.49)	0.51 (+0.28)	0.43 (+0.20)
M-100	0.12	0.69 (+0.57)	0.57 (+0.45)	0.53 (+0.41)	0.39 (+0.27)
M-200	0.26	0.55 (+0.29)	0.54 (+0.28)	0.53 (+0.27)	0.24 (-0.02)
M-400	0.23	0.45 (+0.22)	0.41 (+0.18)	0.43 (+0.20)	0.52 (+0.29)

Table 8: upb

	avman	b1	b2	avsys	tno
M-050	0.97	0.74 (-0.23)	0.64 (-0.33)	0.77 (-0.20)	0.34 (-0.63)
M-100	1.01	0.73 (-0.28)	0.62 (-0.39)	0.78 (-0.23)	0.48 (-0.53)
M-200	0.94	0.72 (-0.22)	0.60 (-0.34)	0.71 (-0.23)	0.52 (-0.42)
M-400	0.95	0.73 (-0.22)	0.43 (-0.52)	0.70 (-0.25)	0.59 (-0.36)

Table 9: ur

	avman	b1	b2	avsys	tno
M-050	0.57	0.25 (-0.32)	0.31 (-0.26)	0.38 (-0.19)	0.46 (-0.11)
M-100	0.59	0.30 (-0.29)	0.48 (-0.11)	0.40 (-0.19)	0.50 (-0.09)
M-200	0.60	0.43 (-0.17)	0.50 (-0.10)	0.46 (-0.14)	0.58 (-0.02)
M-400	0.59	0.33 (-0.26)	0.61 (+0.02)	0.50 (-0.09)	0.58 (-0.01)

Table 10: pp

	avman	b1	b2	avsys	tno
M-050	0.48	0.22 (-0.26)	0.28 (-0.20)	0.27 (-0.21)	0.19 (-0.29)
M-100	0.55	0.17 (-0.38)	0.29 (-0.26)	0.28 (-0.27)	0.27 (-0.28)
M-200	0.57	0.19 (-0.38)	0.32 (-0.25)	0.32 (-0.25)	0.39 (-0.18)
M-400	0.53	0.18 (-0.35)	0.34 (-0.19)	0.34 (-0.19)	0.37 (-0.16)

Table 11: pr

	avman	b1	b2	avsys	tno
M-050	1.02	0.37 (-0.65)	0.46 (-0.56)	0.50 (-0.52)	0.33 (-0.69)
M-100	1.17	0.33 (-0.84)	0.61 (-0.56)	0.55 (-0.62)	0.52 (-0.65)
M-200	1.41	0.38 (-1.03)	0.68 (-0.73)	0.68 (-0.73)	0.82 (-0.59)
M-400	1.35	0.41 (-0.94)	0.77 (-0.58)	0.77 (-0.58)	0.85 (-0.50)

Table 12: pc

measure	avman	b1	avsys	tno
opg	3.84	3.24(-0.60)	3.61(-0.23)	3.67(-0.17)
opc	2.99	2.93(-0.06)	2.65(-0.34)	2.47(-0.52)
opo	3.34	3.09(-0.25)	2.85(-0.49)	2.57(-0.77)
upg	0.61	0.25(-0.36)	0.31(-0.30)	0.26(-0.35)
upm	2.50	2.60(+0.10)	2.59(+0.09)	2.62(+0.12)
upb	0.26	0.50(+0.24)	0.33(+0.07)	0.40(+0.14)
ur	0.91	0.76(-0.15)	0.68(-0.23)	0.56(-0.35)
pp	0.64	0.61(-0.03)	0.62(-0.02)	0.61(-0.03)
pr	0.60	0.45(-0.15)	0.44(-0.16)	0.42(-0.18)
pc	1.65	1.29(-0.36)	1.21(-0.44)	1.10(-0.55)

Table 13: Single document summaries

References

[1] Adam L. Berger and Vibhu O. Mittal. OCELOT: a system for summarizing web pages. In *Proceedings of the 23rd Annnual International ACM SIGIR Conference on research and Development in Information Retrieval*, pages 144–151, 2000.

[2] Jaime G. Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 335–336. ACM, 1998.

[3] H.P. Edmundson. New methods in automatic abstracting. *Journal of the Association for Computing Machinery*, 16(2), 1969.

[4] Djoerd Hiemstra, Wessel Kraaij, Renée Pohlmann, and Thijs Westerveld. Twenty-one at clef-2000: Translation resources, merging strategies and relevance feedback. In Carol Peters, editor, *Proceedings of the CLEF 2000 Cross-Language Text Retrieval System Evaluation Campaign*, 2001, to appear.

[5] W. Kraaij, R. Pohlmann, and D. Hiemstra. Twenty-one at TREC-8: using language technology for information retrieval. In *The Eighth Text Retrieval Conference (TREC-8)*. National Institute for Standards and Technology, 2000.

[6] Julian Kupiec, Jan O. Pedersen, and Francine Chen. A trainable document summarizer. In *SIGIR'95, Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA.*, pages 68–73. ACM Press, 1995.

[7] H. P. Luhn. The automatical creation of literature abstracts. *IBM Journal of Research and Devlopment*, 2(2), 1999.

[8] M. Spitters and W. Kraaij. Using language models for tracking events of interest over time. In *Proceedings of LMIR 2001*, pages 60–65, Pittsburgh, USA, 2001.

Acknowledgements

We thank Michel Simard (RALI, Université de Montréal) for making his sentence splitter available. We thank Horacio Saggion (University of Sheffield), Hap Kolb (TNO TPD), Renée Pohlmann (TNO TPD), Djoerd Hiemstra (University of Twente) and Thijs Westerveld (University of Twente) for their advice.