Writing Style Recognition and Sentence Extraction

Hans van Halteren
Dept. of Language and Speech, Univ. of Nijmegen
hvh@let.kun.nl

Abstract

This paper examines whether feature sets which have been developed for authorship attribution can also be used for the sentence extraction task. Experiments show that the feature sets distinguish significantly better between extract and non-extract sentences than a random baseline classifier, but that a careful combination with other features is necessary in order to outperform a positional baseline classifier. Furthermore, it is vital that the training material reflects the intended task.

1 Introduction

A possible starting strategy for automatic document summarization is sentence extraction (cf. e.g. Mani, 2001). An extraction system attempts to select those sentences from the document which contain the most important information in that document. Ideally, a thorough analysis using linguistic and world knowledge would be brought to bear on the document to determine the appropriate sentences. In most real systems, however, the sentences are selected on the basis of a limited set of much more mundane features, such as sentence length or sentence position within the document. The strategy I present in this paper is also feature-based extraction. Its novelty lies in the machine learning technique used to combine the various features, but also in the actual features that are used, viz. including a number of features which were originally developed to recognize writing style.

The idea underlying this inclusion is that, when we attempt to summarize an article by way of sentence extraction, we assume that the most important information is concentrated in specific sentences. If this is indeed the case, the author of the article must have known where this information was placed. It is therefore possible that he, consciously or subconsciously, wrote these sentences in a different 'style'[1] from the rest of the article. In this paper I examine whether this supposition can lead to valuable additions to the toolbox for sentence extraction.

In Section 2 I present a pilot experiment in which I use the feature sets developed for style recognition (in the context of an authorship attribution task) directly in order to try to distinguish between extract and non-extract sentences. In Section 3 I describe how the apparently most useful features are combined in an extraction system. In Section 4 I show the results of this system in the DUC2002 competition and re-evaluate the choices made during the construction of the extractor in the light of the newly available DUC2002 data. In Section 5 I collect the most important conclusions and describe some future activities.

2 Using style recognition methods for sentence extraction

2.1 Introduction

The automatic recognition of writing styles is studied most notably in the context of the authorship attribution task. In this task one examines a given text and attempts to determine which of a given group of authors has written this text (cf. e.g. Holmes, 1998). The basis for the decision is information about several "style markers", such as vocabulary size or the distribution of a small set of specific vocabulary items. The information about the markers is generally learned from inspection of other texts by the same authors. One of the main focuses of authorship attribution research is the creation of an inventory of useful

[1] The word "style" here should probably not be taken in its literary sense. Rather it refers to measurable recurring patterns in the usage of vocabulary, grammar, text structure, etc.

style markers (cf. Rudman, 1998). Another important focus is the development of techniques with which these style markers can be used to provide sufficiently reliable probability estimates for each potential author (cf. e.g. Baayen et al, 1996).

I my own work on authorship attribution (cf. van Halteren et al, In Prep.), I developed both a new technique to estimate probabilities and sets of features (style markers) which work well with this technique. In this section, I first present the features and the machine learning system, and then describe an experiment which is to show that the system can also be used to locate likely extract sentences in a document.

2.2 Style recognition features

2.2.1 Feature sets for style recognition

The number of features which can conceivably be used for style recognition is enormous. In most cases a limited set of such features is selected, mostly because the systems that have to work with the features can handle only so many. Specific researchers tend to specialize on specific feature sets, partly inspired by the kind of texts they study. In my own authorship attribution work, I have avoided specializing on a single set of features. Instead, I use several different sets, each focusing on specific aspects of the text. In all features sets, the features refer to properties of individual tokens, i.e. single token at a specific position in the text is selected and the features express properties of this token and its context at this position. As an example, take the second occurrence of the token “the” in the sentence “The vice-president of the company had to resign last month.” This token could be assigned properties like ***current=”the”***, ***previous=”of”***, ***part-of-speech=article*** and ***position=4.***

2.2.2 Features in the *trigrams* set

The first feature set, named *trigrams*, is the simplest. It focuses on the lexical context, combined with positional information within the sentence. In this way, it has no need for any linguistic analysis or access to the context beyond the sentence, and it can even be used on very small fragments of text. The actual features are:

1. The ***current token***. The token in its full form, as it occurs in the text, so including capitalization and/or diacritics. Note that punctuation marks are also seen as individual tokens.
2. The ***previous token***. The token immediately to the left of the current token, or a special marker if the current token is the first token in the sentence.
3. The ***next token***. The token immediately to the right of the current token, or a special marker if the current token is the last token in the sentence.
4. The ***sentence length***. The length, in tokens, of the sentence in which the current token is found. The length is mapped onto one of seven possible values: ***1, 2, 3, 4, 5-10, 11-20, 21+.***[2]
5. The ***position within the sentence***. This feature can take three possible values. In sentences of length six or higher, the first three tokens are assigned the value ***Start***, the last three the value ***End*** and the rest the value ***Middle***. In shorter sentences, only ***Start*** and ***End*** are used, the dividing point being the middle of the sentence.

2.2.3 Features in the *tags* set

The second feature set, named *tags*, uses a bit more knowledge about the tokens, in the form of frequency information and wordclass tags. The actual features are:

1. The ***current token***. As above.

[2] For reasons of learnability and model size reduction, numerical values are usually mapped to a small number of ranges. The ranges are generally chosen intuitively rather than on the basis of statistical analysis.

2. The ***wordclass tag for the current token***. The tag is provided by an automatic tagger, which is based on the written texts from the BNC Sampler CDROM.[3]
3. The ***wordclass tag for the previous token***, i.e. for the token immediately to the left of the current token. The tag is replaced by a special marker if the current token is the first token in the sentence.
4. The ***wordclass tag for the next token***, i.e. for the token immediately to the right of the current token. The tag is replaced by a special marker if the current token is the last token in the sentence.
5. The ***frequency*** of the token in the current document. The frequency is mapped onto one of five possible values: ***1, 2-5, 6-10, 11-20, 21+***. Because of the mapping, no further mechanism is used to normalize the frequency on the basis of the document length.
6. The ***sentence length***. As above.
7. The ***position within the sentence***. As above.

2.2.4 Features in the *distribution* set

The third feature set, named *distribution*, ignores the local context, but instead focuses on token distribution within the document. The actual features are:

1. The ***current token***. As above.
2. The ***wordclass tag for the current token***. As above.
3. The ***frequency*** of the token in the current document. As above.
4. The ***token length***. The length of the token, in ASCII characters. The length is mapped onto one of seven possible values: ***1, 2, 3, 4, 5-10, 11-20, 21+.***
5. The ***distribution*** of the token in the document. The document is split into seven equally-sized (in terms of number of tokens) consecutive parts, and the number of blocks in which the current token occurs is counted. This number is then mapped onto one of four possible values: ***1, 2-3, 4-6, 7***.
6. The ***distance to the previous occurrence*** of the token. The distance is measured in sentences and mapped onto one of seven possible values: ***NONE*** (meaning the current occurrence is the first), ***0*** (meaning the previous occurrence is in the same sentence), ***1, 2-3, 4-7, 8-15,16+.*** In order for a token to be recognized as the same token, its form must match exactly, e.g. including capitalization.
7. The ***distance to the next occurrence*** of the token. As the previous feature, but using the distance to the next occurrence of the token.

2.3 Classification software

2.3.1 The WPDV system

The classification software I am using is built around the Weighted Probability Distribution Voting machine learning system (cf. van Halteren, 2000b).

Weighted Probability Distribution Voting (WPDV) is a supervised learning approach to the automatic classification of items. The set of information elements about the item to be classified, generally called a “case”, is represented as a set[4] of feature-value pairs, e.g. the following set from the authorship attribution task using the *trigrams* feature set on the example above:

$$\mathbf{F_{case} = \{ f_{cur}=\text{"the"}, f_{prev}=\text{"of"}, f_{next}=\text{"company"}, f_{len}=5\text{-}10, f_{pos}=\text{middle} \}}$$

The values are always treated as symbolic and atomic, not e.g. numerical or structured, and taken from a finite (although possibly very large) set of possible values. An estimation of the probability of a specific class for the case in question is then based on the number of times that class was observed with those same

[3] A description of the tagger is given by van Halteren (2000a). A description of the corpus and the tagset used on the BNC sampler (C7) is given at http://www.hcu.ox.ac.uk/BNC/getting/sampler.html .

[4] In this section the words “set” and “subset”are used in their mathematical sense.

feature-value pair sets in the training data. To be exact, the probability that class **C** should be assigned to $\mathbf{F_{case}}$ is estimated as a weighted sum over all possible subsets $\mathbf{F_{sub}}$ of $\mathbf{F_{case}}$:

$$\mathbf{P(C) = N(C) \sum_{F_{sub} \subset F_{case}} W_{F_{sub}} \; (freq(C \mid F_{sub}) / freq(F_{sub}))}$$

with the frequencies (**freq**) measured on training data, and **N(C)** a normalizing factor such that

$$\mathbf{\sum_{C} P(C) = 1}$$

The weight factors $\mathbf{W_{F_{sub}}}$ can be assigned in many different ways,[5] but in the current sentence extraction task, there is so little training material that it is simply based on the number of elements in the subset under consideration:

$$\mathbf{W_{F_{sub}} = B^{|Fsub|}}$$

Initial experiments indicate that a weight-base **B** of 0.8 yields good results.

2.3.2 Determination of sentence scores

The WPDV system, then, estimates the probability $\mathbf{P_{token}}$ that a given token in a given context is in a given style. In a two-way authorship attribution task, these would be the styles of authors A and B. In the sentence extraction task, these would be extract style and non-extract style. In the two-way authorship attribution task, the probabilities per token are summed over all the tokens in the text sample; in the sentence extraction task over all tokens in a sentence. In both cases this yields overall scores $\mathbf{P_{overall}}$ for each style **S**:

$$\mathbf{P_{overall}(S) = 1/sentence\_size \; \sum_{token} \; \textit{IF} \; P_{token}(S) > 0.5 \; \textit{THEN} \; (P_{token}(S) - 0.5)^{D} \; \textit{ELSE} \; 0}$$

The factor **D** is used to give more weight to more decisive local scores.

2.4 The pilot experiment

2.4.1 Data

For a controlled pilot experiment to determine the usefulness of the features and system described above for sentence extraction, it is vital that there is manually annotated data in which extract and non-extract sentences are distinguished. John Conroy was so friendly as to provide some material which was annotated by Mary Ellen Okurowski on the basis of per-document summaries in the DUC2001 data (cf. Conroy et al, 2001). The data consists of 147 documents, from 29 document sets. It contains single-document extracts with an average size of about 400 words.

2.4.2 Task

In the pilot experiment I focus on the precision and recall with respect to the extract sentences in the manually annotated data. However, since I want to take all possible extract sizes into account rather than using one or more predefined sizes, I cannot just measure the precision and recall for the extract with a predefined size. Instead I examine precision/recall curves.

[5] E.g., in the authorship attribution task weight factors are assigned per feature, based on the feature's Gain Ratio (a normalized derivative of Information Gain; cf. Quinlan 1986,1993).

Furthermore, I would like to express precision and recall in terms of information content rather than in terms of simply the number of sentences. Not having information about the information content of each sentence, I will approximate the information content (very roughly) by the number of words in the sentence. This means that a sentence of 23 words which is found in both system output and model output counts 23 points towards precision and recall, rather than just 1 for the sentence as a whole.

The relative quality of two precision/recall curves is judged both visually and, more objectively, in terms of a measurement of the surface below the curve.

2.4.3 Experimental setup

For each DUC2001 document set in the data (e.g. d29e), WPDV models for the various feature sets are trained on sentences from all other document sets. These models are then used to select extract sentences from the document set under consideration.

The relative performance of the models is judged on the basis of mutual comparison, but also comparison with two baseline systems. The first baseline makes a random decision on whether or not a sentence is an extract sentence, and can hence be viewed as an absolute lower limit. The corresponding precision/recall curve is marked “random” in Figure 1 and has a below-curve surface of 0.16. A more competitive baseline is based on one of the most informative features known for the extraction task: it gives preference to sentences which are nearer the start of the document. Its curve is marked “position” in Figure 1 and has a below-curve surface of 0.39.

2.5 Results

Figure 1 shows the precision of the selected extract sentences as a function of the recall of all extract sentences in all the 29 sets. The graphs represent the scores for the optimum settings of D, being 2 for *trigrams* and *tags*, and 1 for *distribution*.

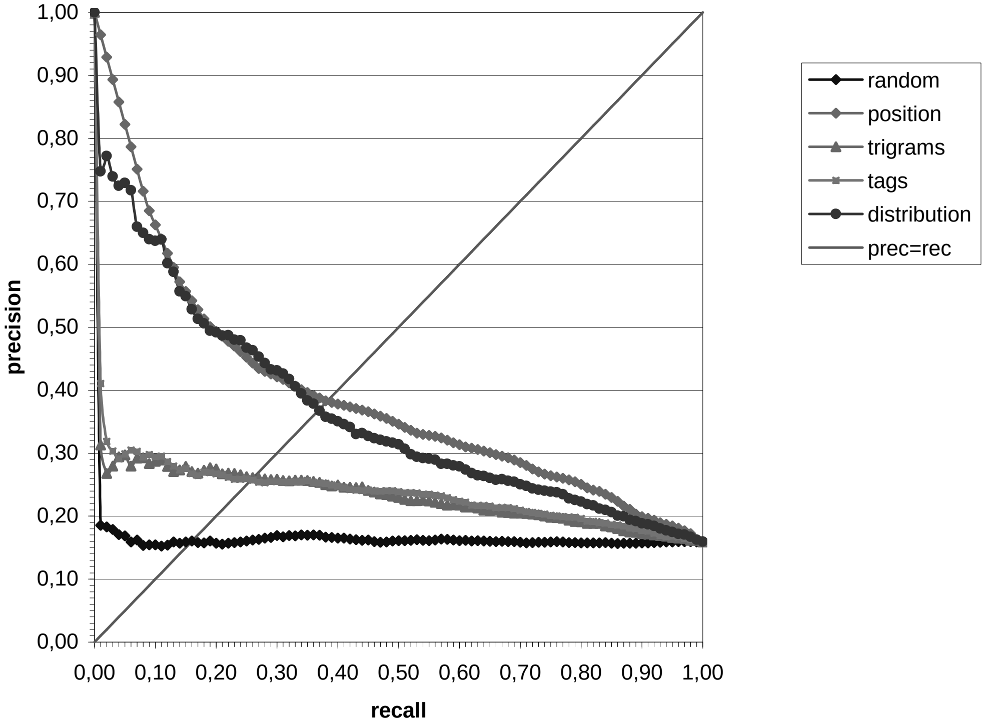

Figure 1: Precision/recall curves (cf. 2.4.2) for various systems. From top to bottom: the positional baseline, the distribution feature set, then two practically overlapping curves: the trigrams and tags feature sets, and finally the random baseline.

The *trigrams* and *tags* feature sets both perform better than chance, with surfaces of 0.23 and 0.24, but are clearly outperformed by the positional system. The *distribution* feature set does much better, with a below-curve surface of 0.36, and is almost as good as the positional system. Since all three have a classification performance well above chance, they could be used to contribute to a feature-based extraction system, but none appears to be strong enough to be used by itself.

3 The WPDV-XTR extraction system

3.1 Introduction

The pilot experiment shows that (parts of) the style recognition feature sets could be a useful addition to the collection of features used in an extraction system. In this section I describe such a system, WPDV-XTR, which uses these new features. Unfortunately, the pilot experiment does not show how the features should be combined exactly with each other and/or with other well-known or new features. Lacking sufficient time for extensive experimentation, I limit myself to observations during the pilot experiment and intuitions about various information sources in the construction of the feature cocktail used in WPDV-XTR.

3.2 Features used

3.2.1 Current token

The current token feature is the same as that used above, in its full form, including capitalization and/or diacritics, and with punctuation marks as separate tokens. My hypothesis is that this feature is probably most useful with function words, which are more likely to be influenced by style than content words. Note that all tokens are used, and not just a selected collection of clear indicators (“cue phrases”) as found in some other systems (cf. e.g. Mani, 2001; Edmundson, 1969).

3.2.2 Distance to previous occurrence

The distance to the previous occurrence of the token is also the same as used above, measured in sentences and mapped onto one of seven possible values: ***NONE*** (meaning the current occurrence is the first), ***0*** (meaning the previous occurrence is in the same sentence), ***1, 2-3, 4-7, 8-15,16+.*** As a closer examination of the pilot experiment results showed, this feature is easily the most informative of all those used so far, and is therefore kept as is.

3.2.3 Sentence length

The feature for sentence length is kept intact in the form described above, mapped onto one of seven possible values: ***1, 2, 3, 4, 5-10, 11-20, 21+***. Although it does not appear to do all that well in the pilot experiment, it is described as useful in the extraction literature (e.g. Kraaij et al 2001; Sekine and Nobata, 2001).

3.2.4 Tag context

The wordclass tags of the tokens in the direct context have proved to be useful, but not useful enough to warrant several separate features. Instead I combine them into a single feature, which consists of either the sequence of three tags ending at the current position or the sequence of three tags starting at the current position. Both forms are used for each token position, but in the same feature position, using feature overloading.[6]

[6] The WPDV system allows the same feature to occur more than once in a single case, with different values. Both feature-value pairs are used in combinations, but they are not combined with each other.

3.2.5 Token distribution

Another type of information which has shown to be useful but needing a more compact form is that about the distribution of the current token within the document and across documents. The feature in its new form is a concatenation of three separate pieces of information:

- The frequency of the token in the current document, as above, mapped onto one of five possible values: ***1, 2-5, 6-10, 11-20, 21+***.
- The presence of the token in 1/7th document blocks, as above, mapped onto one of four possible values: ***1, 2-3, 4-6, 7***.
- The ***document bias*** of the token, which represent the token's tendency to have its occurrences concentrated in a few documents.[7] To calculate the document bias of a token, divide the frequency per thousand words in those documents containing the token by frequency per thousand words over the whole document collection,[8] and take the log of the result. A low document bias indicates a token used predominantly as a function word, a high document bias indicates a token used predominantly as a content word. In these experiments, the document bias is rounded down to one of five possible values: ***0, 1, 2, 3, 4***.

3.2.6 Relative token frequency

Information about the importance of the current token in the document is given by the ***relative frequency*** feature. It is calculated by dividing the frequency per thousand words in the current document by the frequency per thousand words in those documents from the document collection containing the token. Generally, the tokens central to the document's subject will have the highest relative frequency. Here, the relative frequency is mapped onto one of six possible values, each representing a range; ***0-1/2*** (less than 0.5 times expected), ***1/2-3/2*** (between 0.5 and 1.5 times expected), ***3/2-2*** (between (1.5 and 2 times expected), ***2-4*** (2 to 4 times expected), ***4-8*** (4 to 8 times expected) and ***8+*** (more than 8 times expected).

3.2.7 Sentence position

The final feature in the cocktail is the one recognized as one of the best features for the task: the position of the sentence in the document. Since most documents in the sources for the DUC2002 data have no information about paragraph borders, only the absolute position in the document can be used. It is mapped here onto one of six possible values: ***1, 2-3, 4-7, 8-15, 16-31, 32+***.

[7] Document bias is similar to the better-known inverse document freqeuncy (IDF). They differ mostly in that document bias takes document length into account.

[8] The document collection referred to consists of all documents on the TIPSTER and TREC CDROMs from the same source as the current document, e.g. all WSJ documents.

3.3 Results for training data

On the training data, WPDV-XTR (with the decisiveness factor D set to 4) performs better than the positional baseline on the chosen criterion, with a below-curve surface of 0.42. However, as can be seen in Figure 2, at the high precision/low recall end of the curve, it performs worse than the baseline. This may mean that WPDV-XTR might yet perform worse than the positional baseline when having to produce small extracts.

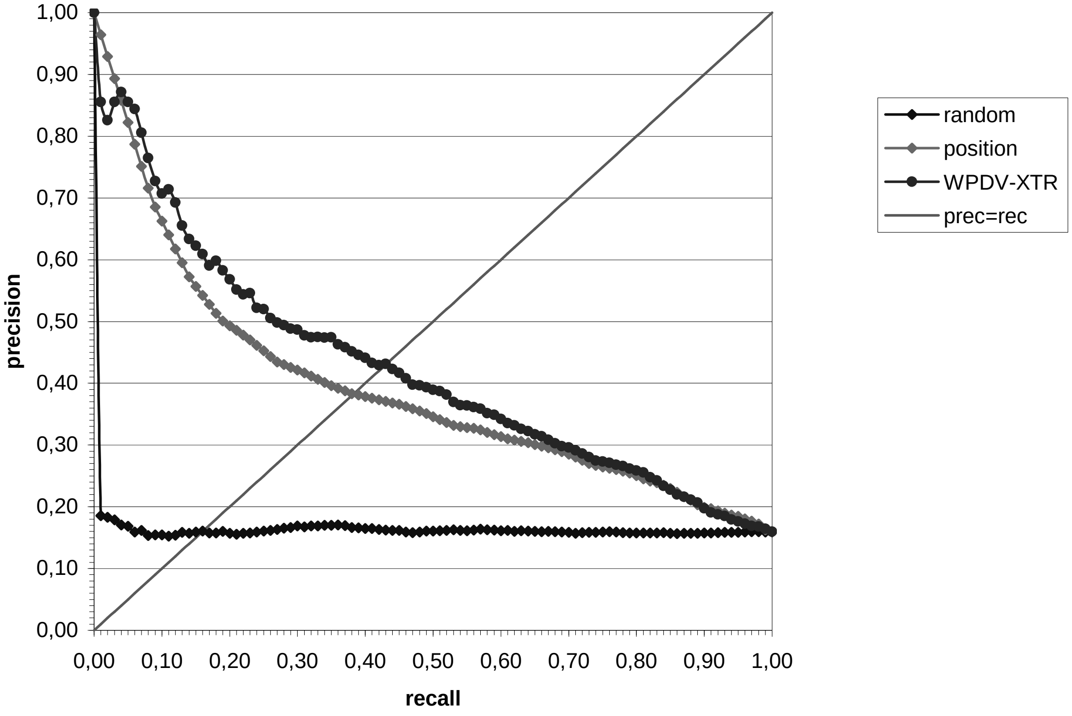

Figure 2: Precision/recall curves (cf. 2.4.2) for WPDV-XTR and two baseline systems. From top to bottom: WPDV-XTR, the positional baseline and the random baseline.

4 The DUC2002 competition and a re-evaluation of system settings

4.1 Introduction

To measure the performance of WPDV-XTR with its style recognition features against other state-of-the-art extraction systems, I participated in the DUC2002 competition, as described on the webpages at http://www-nlpir.nist.gov/projects/duc/2002.html. In the numerical coding system provided by the DUC2002 organization for anonymity of the systems, WPDV-XTR is coded number 21; in this paper I use its name instead.

The extraction part of the competition (multi-document extraction) consisted of the creation of two generic sentence extracts of the documents sets of approximately ten newswire/newspaper documents (articles) on a single subject. The extracts were to form an extract of the whole document set in approximately 400 and 200 (whitespace-delimited) tokens. The extracts had to consist of some subset of the "sentences" predefined by NIST in a sentence-separated form of the document set and each predefined sentence had to be used in its entirety or not at all in constructing an extract.

In the end, ten of the groups participating in DUC2002 submitted multi-document extracts.

4.2 Evaluation

As I am concentrating on precision and recall in this paper, I will keep using these measurements. However, I now switch from the overall curve to two specific sampling points: the 200-word and the 400-word extract as defined in the DUC2002 task.

For these two points I will calculate precision and recall with regard to the human summarizers' extracts. In addition to the precision and recall with consideration of sentence length, as used above, I will now also calculate precision and recall purely in terms of sentences. The truth with regard to information content should lie somewhere in the middle.

To derive a single measurement, I combine precision and recall into an F-value (van Rijsbergen, 1975):

$$\mathbf{F = ((\beta^2+1)*precision*recall) / (\beta^2*precision + recall)}$$

with β equal to 1:

$$\mathbf{F_{\beta=1} = (2*precision*recall) / (precision+recall)}$$

To get an overall quality assessment I average the F-value over all measured extracts. Ideally there would have been 60 such extracts, but due to administrative mix-ups one document collection (d088) received no human summarizers' extracts and two collections (d076 and d098) received only one. To keep the measurements balanced I restrict the evaluation to the remaining 57 collections. Given the number of sentences under consideration and the values in question, the 95% confidence intervals for all F-values for 200-word extracts are about ±0.035 and for 400-word extracts about ±0.030.

The restriction to those collections for which there are two human summarizers' extracts also allows a comparison of the results for automatic extraction with those for manual extraction. The average F-values for inter-summarizer measurements of 0.250 (S) / 0.270 (W) for the 400-word extracts and 0.213 (S) / 0.222 (W) for the 200-word extracts are rather low. This means that we will not be able to draw very strong conclusions, as it is unclear whether one or two manually created extracts form a sufficient basis for a good benchmark.

In the discussion below I will again be using positional baseline systems which select sentences closer to the start of the documents. There are two such systems. The first, named TBASE, has been provided to me by Wessel Kraaij and his group at TNO-TPD. It uses the sentence split as provided in the DUC2002 data. It selects the first sentence of each document in the set, then each second sentence, etc. until the length of the extract is between 95% and 100% of the required length. The second system, WBASE, differs mostly in that it uses the sentences as perceived by my own tokenizer rather than those provided by DUC. Another difference is that it aims for extracts at least the required length and at most 10 words over. Given that WPDV-XTR and its variants below all use my own tokenizer, WBASE is my reference point, and my quality target. For the other systems, TBASE is the better reference point.

Apart from the positional baselines, I also include the comparison of the two manual extracts for each document collection and measurements on a random baseline extractor.[9]

4.3 Submission system results

The F-values for the various systems and the positional baselines for 200-word extracts are as follows:

[9] The random baseline is actually the average of three runs of a random baseline system using different seeds for the random number generator.

System	Sentence-based F-value	Word-based F-value
MAN-MAN	0.213	0.222
WBASE	0.198	0.219
TBASE	0.191	0.215
WPDV-XTR	***0.188***	***0.211***
SYS19	0.183	0.199
SYS24	0.172	0.193
SYS28	0.136	0.167
SYS20	0.126	0.144
SYS29	0.089	0.102
SYS31	0.082	0.094
SYS25	0.080	0.092
SYS16	0.063	0.077
SYS22	0.038	0.042
RANDOM	0.030	0.032

Given the confidence intervals, it would seem WPDV-XTR, system 19 and system 24 form a leader group, with no significant differences amongst each other, but significantly better than the rest. Their extracts are also not significantly worse than the manual ones, except for system 24 when measured at the sentence level.

For 400-word extracts, the results are:

System	Sentence-based F-value	Word-based F-value
WBASE	0.269	0.299
TBASE	0.265	0.294
WPDV-XTR	***0.258***	***0.290***
MAN-MAN	0.250	0.270
SYS19	0.223	0.240
SYS24	0.222	0.249
SYS28	0.197	0.241
SYS20	0.172	0.191
SYS29	0.156	0.179
SYS31	0.153	0.172
SYS25	0.148	0.165
SYS16	0.128	0.156
SYS22	0.084	0.097
RANDOM	0.070	0.078

Here, WPDV-XTR performs significantly better than the other submissions systems (with the given method of measuring), and even better than the human summarizers, but still worse than the positional baseline(s).

4.4 Re-evaluation

The question, now, is whether WPDV-XTR cannot improve upon the positional baseline for this task in principle, or whether some of the choices made within the system have been the wrong ones. To investigate this question, I ran additional experiments, to test if variations of the system can improve on the positional baseline. In the subsections below I present the results for the experiments in increasing degree of variation: changes in parameter settings, changes in feature selection and switching the training data to the DUC2002 extracts. Note that these results greatly profit from hindsight, as I can now make the optimal choices with regard to the test data; these experiments should therefore be seen mostly as an investigation of where the ceiling is for this task for the WPDV-XTR system.

4.5 Parameter settings

First of all, it is possible to run the system with different parameter settings, the parameters being the weight-base factor B used in WPDV feature combination and the decisiveness factor D. In the WPDV-XTR system as participating in DUC2002 these had been set to 0.8 and 4 respectively. These values were chosen without thorough experimentation, however, due to a rapidly nearing DUC2002 deadline. New experiments show that these settings are not optimal, but not significantly worse than the observed optimal settings either.

For the 200-word extract, the F-values for the various settings range from 0.179 (S) / 0.198 (W) to 0.205 (S) / 0.224 (W), none significantly worse or better than the 0.188 (S) / 0.211 (W) of the participating system. The highest values appear to be found for lower weight-bases and higher decisiveness factors, with 0.205/0.224 at B=2, D=7. This would indicate that, for the 200-word extraction task, individual clues are more important than combinations of clues (because low B does best) and only the most decisive clues should be taken into account (because high D does best). A possible explanation for this is that the task at hand is different from the single-document extraction task: in the 200-word multi-document extraction task, it is not so much the sentences within a document that are competing for a place in the extract, but rather the high-scoring sentences from different documents. The new high values are also higher than those for WBASE, 0.198 (S) / 0.219 (W).

For the 400-word extract, the F-values for the various settings range from 0.245 (S) / 0.272 (W) to 0.262 (S) / 0.294 (W), none significantly worse or better than the 0.258 (S) / 0.290 (W) of the participating system. Here, the highest scores are much less localized. They are found both for lower B with D=7, and for slightly higher B and D=5. No setting leads to a better score than that of WBASE, viz. 0.269 (S) / 0.299 (W).

So for both extract sizes, the settings selected for the submission are not optimal, but only for the 200-word extract can the optimization of the settings lead to an improvement over the positional baseline.

4.6 Feature selection

Another thing which can be varied is the set of ingredients of the feature cocktail. In this section, I will examine two variations only:

- I measure the effectiveness of single features by themselves by running the system with each case containing only that feature.
- I leave single features out of the cocktail and measure each contribution as final ingredient by subtracting the performance by the cocktail without the feature from the performance of the cocktail with the feature. Note that the contribution may be negative, which means that removing the feature actually improves performance.

For the 200-word extracts the individual features' effectiveness is:

Feature	**Contribution S/W**	**Single S/W**
sentence position	+0.054/+0.062	0.202/0.227
distance to previous	+0.010/+0.012	0.159/0.180
current token	–0.008/–0.007	0.058/0.066
relative token frequency	–0.015/–0.015	0.051/0.059
token distribution	–0.002/–0.001	0.048/0.054
tag context	–0.005/–0.004	0.049/0.052
sentence length	–0.023/–0.019	0.031/0.036

and for the 400-word extracts:

Feature	**Contribution S/W**	**Single S/W**
sentence position	+0.068/+0.076	0.273/0.300
distance to previous	+0.016/+0.026	0.246/0.277
relative token frequency	+0.005/+0.015	0.108/0.129
token distribution	+0.002/+0.012	0.110/0.127
current token	+0.006/+0.018	0.110/0.126
tag context	–0.011/–0.000	0.093/0.102
sentence length	–0.012/–0.001	0.077/0.102

In both cases, sentence position is indispensable and distance to previous occurrence very useful. There are also features which appear to be a hindrance rather than an asset, although none of the variants perform significantly worse than the participating system. However, it should also be noted that I have measured only single removals. That a feature has no valuable contribution as a final ingredient does not mean it has no contribution at all. But it would certainly seem that sentence length (which itself performs hardly above chance level, about 0.030/0.032, cf. 4.3) can be removed when creating 200-word extracts.

4.7 Training data

The biggest change is the replacement of the training data used for the WPDV-XTR as participating (400-word single document extracts) by the DUC2002 extracts (200- and 400-word multi-document extracts). For each collection, I retrain the system on all other collections and create a new submission for the collection in question. This experiment uses reasonably optimal conditions. The training data is of the exact same type as the test data, being selected for the same purpose and having been given extracts by the same group of extractors. However, as the training and the test data have been carefully separated, the experiment is still fair.[10]

Even without any changes in feature selection or parameter settings, the results of the experiment are striking:[11]

	F-value 200-word S/W	**F-value 400-word S/W**
WPDV-XTR	0.188/0.211	0.258/0.290
WBASE	0.198/0.219	0.269/0.299
Retrained on 200-word	***0.217/0.237***	0.276/0.308
Retrained on 400-word	0.207/0.230	***0.277/0.309***

[10] But note that there are instances of the same document occurring in more than one collection (30 documents in 2 collections and 2 documents in 3 collections). In such cases some information leakage might occur. However, a closer examination shows that the documents receive rather different extracts in different collections. Take the 200-word manual extracts. In the 66 extracts for repeated documents, we find 254 words (tokens, not types) that are included in both manual extracts for one of the collections in which the document occurs. Of these 254 words, only 124 are also included in both extracts for another collection, 94 in just one extract for another collection, and 136 not at all in the extracts for the other collection. We also find 1754 words in these 66 extracts that are included in just one extract for one collection. Here, reuse is even lower with only 100 words included in both extracts for another collection, 496 in just one, and 1158 not at all. It seems like prior knowledge about the document is misleading at least as often as it is helpful.

[11] In fact, one parameter was changed. For reasons of computation time, the WPDV models for this retraining experiment are based only on those features and combinations that occur at least two times in the training data. This generally leads to a loss of quality, so the results are slightly worse than they could have been with a full model.

For both extract sizes, retraining lifts the performance above that of the positional baseline and of all examined variants above (although again not significantly). It would seem the disappointing performance of the DUC2002 submission is primarily caused by the discrepancy between the kind of training data and the intended task.

4.8 Combined changes

Given that retraining, changes in feature selection and parameter optimization all lead to improved results, it stands to reason that a combination of the three brings an even better improvement. Here I will limit myself to a single clear example of this effect: a system trained on the 200-word DUC2002 extracts as described above, but without the frequency threshold (cf. footnote 11), and leaving out the features for sentence length and relative token frequency.

The system does quite well for both extracts sizes, although at completely different parameter settings. For the 200-word extracts, the best settings are where we expect them by now, at high D and lower weights. But for the 400-word extracts, the lower D suddenly do well, and even high weights start bringing results.

The new best score for 400-word extracts is 0.287 (S) / 0.317 (W), which is not a significant improvement and a mere 6% over the positional baseline, but then the model is trained for 200-word extracts. At its intended extract size, 200, the new best score, 0.245 (S) / 0.265 (W), is an impressive and statistically significant improvement, a full 24% (S) / 21% (W) better than the positional baseline.

5 Conclusion

The experiments described in this paper show that WPDV-XTR, using features based on writing style recognition, is able to produce extracts which are on a par with, or better than other state-of the-art systems. The system is also able to perform significantly better than positional baselines, but only if it can be tuned to the task at hand. Factors in the tuning process are, in order of importance:

1. appropriateness of the training data to the task
2. feature selection
3. parameter settings

The appropriateness of the training data is easily the most important. I expect that further improvements for the machine learning approach to multi-document sentence extraction is possible with even more specialized training data. The lack in overlap between the extracts for the same collection by different human summarizers shows that there is at most a small overlapping extract segment, but outside that each summarizer makes his own particular choices, possibly on the basis of his own interpretation of the instructions by the DUC2002 organization (which do not specify any goal or focus for the extracts). In other words, each summarizer brings his own bias to the extraction task. The ability of the machine learning system to reach a larger overlap with the summarizers than they have among themselves appears to indicate that the system imitates the specific choices, the biases of the human summarizers. If this is true it would probably be best for the creation of future training data to:

- give more exact instructions about the desired goal/focus of the extracts, so that biases are prevented or at least lessened
- limit the number of human summarizers, possibly even to one, to prevent biases or extract style differences other than the goal- or focus-directed ones

With such training data, extracting should be much easier to learn than the current mix of different biases.

Once the training material and the task have been selected, feature selection and parameter settings can be optimized. The experiments have shown that not much can be taken for granted here, as optimal choices vary with changes in the task (e.g. 200-word versus 400-word extracts) and even features which are generally assumed to be useful, e.g. sentence length, can prove less than useful under some circumstances.

All in all, WPDV-XTR appears to provide a sufficient basis for future research. However, such research will have another, and much wider, focus. The current experiments were aimed at the DUC2002 competition and its defined criteria, i.e. precision and recall with regard to manual extracts. For actual applications, rather different criteria have to be, and can be, optimized. When using extracts as a basis for abstracts, recall is probably more important than precision as the subsequent transformations can further reduce the abstract length. When using extracts in a question-answering system, a question is available to give direction to the extracting process. Whatever the chosen task, a purely quantitative evaluation should also be complemented by more qualitative evaluations, so that the most troublesome shortcomings of the system can be tracked down and repaired. Furthermore, for such repairs, and for the larger task in which the extractor is embedded, it will probably be necessary to look beyond the pure machine learning approach and include more linguistically motivated techniques.

References

Baayen, R.Harald, Hans van Halteren and Fiona Tweedie. 1996. Outside the Cave of Shadows: Using syntactic annotation to enhance authorship attribution. Literary and Linguistic Computing 7, 91-109.

Conroy, John M., Judith D. Schlesinger, Dianne P. O'Leary and Mary Ellen Okurowski. 2001. Using HMM and Logistic Regression to Generate Extract Summaries for DUC. Proc. DUC2001.

Edmundson, H.P. 1969. New methods in automatic abstracting. Communications of the Association for Computing Machinery 16(2): 264-285.

van Halteren, Hans. 2000a. The detection of inconsistency in manually tagged text. Proc. LINC2000.

van Halteren, Hans. 2000b. A default first order weight determination procedure for WPDV models. Proc. CoNNL 2001, pp 119-122.

van Halteren, Hans, R. Harald Baayen, Fiona Tweedie and Anneke Neijt. In Preparation. New Machine Learning Methods Support the Theorized Existence of a Human Stylome.

Holmes, David. 1998. Authorship attribution. Literary and Linguistic Computing 13(3), 111-117.

Kraaij, Wessel, Martijn Spitters and Martine van der Heijden. 2001. Combining a mixture language model and Naive Bayes for multi-document summarization. Proc. DUC2001.

Mani, Inderjeet. 2001. Automatic summarization. Amsterdam/Philadelphia: John Benjamins Publishing Company.

van Rijsbergen, C.J. 1975. Information Retrieval. Buttersworth.

Rudman, Joseph. 1998. The state of authorship attribution studies: some problems and solutions. Computers and the Humanities 31: 351-365.

Sekine, Satoshi and Chikashi Nobata. 2001. Sentence extraction with information extraction technique. Proc. DUC2001.