Ben Hachey and Claire Grover, 'Sentence Classification Experiments for Legal Text Summarisation' in T. Gordon (ed.), *Legal Knowledge and Information Systems. Jurix 2004: The Seventeenth Annual Conference*. Amsterdam: IOS Press, 2004, pp. 29-38.

Sentence Classification Experiments for Legal Text Summarisation

Ben Hachey and Claire Grover
School of Informatics, University of Edinburgh
{bhachey,grover}@inf.ed.ac.uk

Abstract. We describe experiments in building a classifier which determines the rhetorical status of sentences. The research is part of a text summarisation project for the legal domain and we use a newly compiled and annotated corpus of judgments of the UK House of Lords. Rhetorical role classification is an initial step which provides input to the sentence selection component of the system. We report results from experiments with four classifiers from the *Weka* package (C4.5, naïve Bayes, Winnow and SVMs). We also report results using maximum entropy models both in a standard classification framework and in a sequence labelling framework. The SVM classifier and the maximum entropy sequence tagger yield the most promising results.

1 Introduction

In the SUM project we are building a system for summarising legal judgments that is generic and portable while maintaining a mechanism to account for the rhetorical structure of the argumentation of a case. The importance of summarisation in the legal domain stems from the role that precedents play in common law. Already, a number of content providers are providing access to manual summarisations of legal judgments. An automatic system would enable immediate access to preliminary summaries, and serve as an assisting technology in manual summarisation. Automatic summaries might also be incorporated to provide dynamic, customised content in information retrieval systems.

For example, consider a case database where the user queries using key words or natural language and gets back a list of summaries of possible precedent-setting rulings including an indication of the decision. Alternatively, the whole document could be treated as a query in which case a system could actively search for and summarise documents similar to that which the user is currently viewing. These kinds of systems have great utility both for learning law and especially as a research aid for lawyers.

In Section 2 we describe the approach we are taking. We also describe the corpus of legal judgments that we have gathered and the manual annotation of rhetorical role classification that we have performed. In Section 3 we describe the XML-based automatic linguistic analysis of the corpus which provides features for the classifier. Section 4 contains an overview of the feature sets that we use for our experiments and the results from our experiments with four *Weka* classifiers. In Section 5 we report results for a maximum entropy classifier before investigating treating the task as a sequence labelling problem. Finally, in Section 6, we draw conclusions and outline directions for future work.

2 Background

2.1 Sentence Extraction with Rhetorical Status Information

In this paper we report on a set of experiments to classify sentences for rhetorical status using a wide range of machine learning techniques. The task of classifying sentences forms part of a sentence extraction-based automatic summarisation system in the legal domain. The experiments described are part of an ongoing endeavour to determine the best classification techniques and the best feature sets for the task.

In the SUM project we are exploring methods for generating flexible summaries of legal documents. Our approach to summarisation is described in detail in [11] and takes as a point of departure the work of Teufel and Moens [28, 26, 27] (henceforth T&M). We have chosen to work with law reports in part because the existence of manual summaries means that we will have evaluation material for the final summarisation system.

The T&M approach is an instance of what is known as the *text extraction* method of summarisation. In this approach a summary typically consists of sentences selected from the source text, with some smoothing (e.g reordering, anaphora resolution) to increase the coherence between them. Following T&M, we go beyond simple sentence selection and classify source sentences according to their rhetorical status (e.g. a description of background facts in the case, a reference to a point of law, etc.). With sentences classified in this manner, different kinds of summaries can be generated with prominence given to particular kinds of sentence.

2.2 The HOLJ Corpus

We have gathered a corpus of judgments of the House of Lords (the HOLJ corpus).[1] Each document contains a header providing structured information (e.g., respondent, appellant, date of hearing), followed by a sequence of (usually five) Law Lords' judgments consisting of free-running text, at least one of which is a substantial speech. Typically this will start with a statement of how the case came before the court, move on to a recapitulation of the facts, discuss one or more points of law, and then offer a ruling.

The corpus consists of 188 documents from the years 2001–2003. For 153 of these, manually created summaries are available and will be used for system evaluation.[2] The total number of words in the free text parts of the corpus is 2,887,037 and the total number of sentences is 98,645. The average sentence length is approximately 29 words. A document contains an average of 525 sentences while an individual Law Lord's judgment contains an average of 105 sentences.

The raw HTML documents are processed through a sequence of modules which convert to XML and add layers of linguistic annotation (see Section 3); an individual Law Lord's judgment is encoded as a LORD element. All annotation is computed automatically except for manual annotation of sentences for their rhetorical status. The human annotation of rhetorical roles is performed on the documents after the tokenisation component has performed sentence boundary disambiguation. This annotation is work in progress and so far we have around 70 manually annotated documents. The experiments reported here were conducted using 40 of these. This subset is similar in size to the corpus of 80 academic papers reported in Teufel and Moens [28]. Our corpus contains 290,793 words and 10,169 sentences while the T&M corpus contains 285,934 words and 12,188 sentences. Note that although our corpus contains marginally more words, the T&M corpus has a shorter average sentence length and thus contains more sentences.

[1] `http://www.parliament.uk/judicial_work/judicial_work.cfm`
[2] `http://www.lawreports.co.uk/`

Label	Freq.	Description
FACT	862 (8.5%)	Recounts the events or circumstances giving to legal proceedings. E.g. *On analysis the package was found to contain 152 milligrams of heroin at 100% purity.*
PROCEEDINGS	2434 (24%)	Describes legal proceedings taken in the lower courts. E.g. *After hearing much evidence, Her Honour Judge Sander made findings of fact on 1 November 2000.*
BACKGROUND	2813 (27.5%)	Direct quotation or citation of source of law material. E.g. *Article 5 provides in paragraph 1 that a group of producers may apply for registration ...*
FRAMING	2309 (23%)	Part of the Law Lord's argumentation. E.g. *In my opinion, however, the present case cannot be brought within the principle applied by the majority in the Wells case.*
DISPOSAL	935 (9%)	Credits or discredits a claim or previous ruling. E.g. *I would allow the appeal and restore the order of the Divisional Court.*
TEXTUAL	768 (7.5%)	Has to do with the structure of the document or with things unrelated to a case. E.g. *First, I should refer to the facts that have given rise to this litigation.*
OTHER	48 (0.5%)	Does not fit any of the above categories. E.g. *Here, as a matter of legal policy, the position seems to me straightforward*

Table 1: Rhetorical Annotation Scheme for Legal Judgments

The rhetorical roles that it is appropriate to assign to sentences vary from domain to domain and reflect the argumentative structure of the texts. Teufel and Moens [28] describe a set of labels which reflect regularities in the argumentative structure of research articles following from the author's communicative goals. For scientific articles the role labels reflect such things as the the goals of the paper, sentences describing generally accepted scientific background, etc. For our legal domain, the author's primary communicative goal is to convince his peers that his position is legally sound, having considered the case with regard to all relevant points of law. We have analysed the structure of typical documents in our domain and derived from this seven rhetorical role categories, illustrated in Table 1. The second column shows the frequency of occurrence of each label in the manually annotated subset of the corpus. Apart from the OTHER category, the most infrequently assigned category is TEXTUAL while the most frequent is BACKGROUND. In general, the distribution across categories is more uniform than that of the T&M labels: Teufel and Moens [28] report that their most frequent category (OWN) is assigned to 67% of sentences in their corpus while three other labels (BASIS, TEXTUAL and AIM) are each assigned to only 2% of sentences.

The 40 documents in our manually annotated subset were annotated by two annotators using guidelines which were developed by one of the authors, one of the annotators and a law professional. Eleven files were doubly annotated in order to measure inter-annotator agreement. We used the kappa coefficient of agreement as a measure of reliability. This showed that the human annotators distinguish the seven categories with a reproducibility of K=*.83* (N=1,955, k=2; where K is the kappa co-efficient, N is the number of sentences and k is the number of annotators). This is slightly higher than that reported by T&M and above the .80 mark which Krippendorf [13] suggests is the cut-off for good reliability.

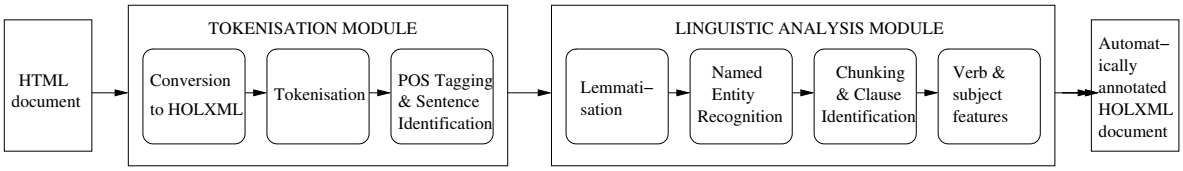

Figure 1: HOLJ Processing Stages

3 Linguistic Analysis

One of the goals of the SUM project is to create an annotated corpus in the legal domain which will be available to other researchers. With this aim in mind we have used the HOLXML format for the corpus and we encode all the results of linguistic processing as XML annotations. Figure 1 shows the broad details of the automatic processing that we perform, with the processing divided into an initial tokenisation module and a later linguistic annotation module. The architecture of our system is one where a range of NLP tools is used in a modular, pipelined way to add linguistic knowledge to the XML document markup. The motivation for the module that performs further linguistic analysis is to compute information to be used to provide features for the sentence classifier. However, the information we compute is general purpose, making the data useful for a range of research activities.

In the tokenisation module we convert from the source HTML to HOLXML and then pass the data through a sequence of calls to a variety of XML-based tools from the LT TTT and LT XML toolsets [9, 29]. These include *fsgmatch*, a general purpose transducer which processes an input stream and adds annotations using hand-written rules and *ltpos*, a statistical combined part-of-speech (POS) tagger and sentence boundary disambiguation module [18].

The first step in the linguistic analysis module lemmatises the inflected words using Minnen et al.'s [19] *morpha* lemmatiser. The next stage, described in Figure 1 as Named Entity Recognition, is in fact a more complex layering of two kinds of named entity recognition. The documents in our domain contain standard kinds of entities such as person, organisation, location and date. However, they also contain entities which are are specific to the domain. Table 2 shows examples of the entities we have marked up in the corpus (noun groups (NG) with specific type and subtype attributes). The table shows both domain-specific entities such as courts, judges, acts and judgments and non-domain-specific entity types. To identify the domain-specific ones we use hand-crafted LT TTT rules, while for the non-domain-specific ones we use the C&C named entity tagger [4] trained on the MUC7 [1] data set.

The next stage in the linguistic analysis module performs noun group and verb group chunking using *fsgmatch* with specialised hand-written rule sets. The noun group and verb group mark-up plus POS tags provide the relevant features for the next processing step. In [8] we showed that information about the main verb group of the sentence is likely to provide clues as to rhetorical status (e.g. present tense tends to correlate more highly with BACKGROUND). In order to find the main verb group of a sentence, however, we need to establish its clause structure. We do this with a maximum entropy clause identifier [10] built using the CoNLL-2001 shared task data [25].

The final stages of linguistic processing use hand-written LT TTT components to compute features of verb and noun groups. For all verb groups, attributes encoding tense, aspect, modality and negation are added to the mark-up: for example, *might not have been brought* is analysed as <VG tense='pres', aspect='perf', voice='pass', modal='yes', neg='yes'>.

In addition, subject noun groups are identified and lemma information from the head noun of the subject and the head verb of the verb group are propagated to the verb group attribute list.

type='enamex-pers' subtype='committee-lord'	*Lord Rodger of Earlsferry, Lord Hutton*
type='caseent' subtype='appellant'	*Northern Ireland Human Rights Commission*
type='caseent' subtype='respondent'	*URATEMP VENTURES LIMITED*
type='enamex-pers' subtype='judge'	*Collins J, Potter and Hale LJJ*
type='enamex-org' subtype='court'	*European Court of Justice, Bristol County Court*
type='legal-ent' subtype='act'	*Value Added Tax Act 1994, Adoption Act 1976*
type='legal-ent' subtype='section'	*section 18(1)(a), para 3.1*
type='legal-ent' subtype='judgment'	*Turner J [1996] STC 1469*
type='enamex-loc' subtype='fromCC'	*Oakdene Road, Kuwait Airport*
type='enamex-pers' subtype='fromCC'	*Irfan Choudhry, John MacDermott*
type='enamex-org' subtype='fromCC'	*Powergen, Grayan Building Services Ltd*

Table 2: Named Entities in the Corpus

4 Initial Classification Experiments

4.1 Feature Sets

The feature set described in Teufel and Moens [28] includes many of the features which are typically used in sentence extraction approaches to automatic summarisation as well as certain other features developed specifically for rhetorical role classification. Briefly, the T&M feature set includes such features as: location of a sentence within the document and its subsections and paragraphs; sentence length; whether the sentence contains words from the title; whether it contains significant terms as determined by the information retrieval metric *tf***idf*; whether it contains a citation; linguistic features of the first finite verb; and cue phrases (described as meta-discourse features in [28]). The features that we have been experimenting with for the HOLJ corpus are broadly similar to those used by T&M.

Location. For sentence extraction in the news domain, sentence location is an important feature and, though it is less dominant for T&M's scientific article domain, they did find it to be a useful indicator. T&M calculate the position of a sentence relative to segments of the document as well as sections and paragraphs. In our system, location is calculated relative to the containing paragraph and LORD element and is encoded in six integer-valued features: paragraph number after the beginning of the LORD element, paragraph number before the end of the LORD, sentence number after the beginning of the LORD element, sentence number before the end of the LORD, sentence number after the beginning of the paragraph, and sentence number before the end of the paragraph.

Thematic Words. This feature is intended to capture the extent to which a sentence contains terms which are significant, or thematic, in the document. The thematic strength of a sentence is calculated as a function of the *tf***idf* measure on words (*tf*='term frequency', *idf*='inverse document frequency'): words which occur frequently in the document but rarely in the corpus as a whole have a high *tf***idf* score. The thematic words feature in Teufel and Moens [28] records whether a sentence contains one or more of the 18 highest scoring words. In our system we summarise the thematic content of a sentence with a real-valued thematic sentence feature, whose value is the average *tf***idf* score of the sentence's terms.

Sentence Length. In T&M, this feature describes sentences as short or long depending

on whether they are less than or more than twelve words in length. We implement an integer-valued sentence length feature which is a count of the number of tokens in the sentence.

Quotation. This feature, which does not have a direct counterpart in T&M, encodes the percentage of sentence tokens inside an in-line quote and whether or not the sentence is inside a block quote.

Entities. T&M do not incorporate full-scale Named Entity Recognition in their system, though they do have a feature reflecting the presence or absence of citations. We recognise a wide range of named entities and generate binary-valued entity type features which take the value 0 or 1 indicating the presence or absence of a particular entity type in the sentence.

Cue Phrases. The term 'cue phrase' covers the kinds of stock phrases which are frequently good indicators of rhetorical status (e.g. phrases such as *The aim of this study* in the scientific article domain and *It seems to me that* in the HOLJ domain). T&M invested a considerable amount of effort in compiling lists of such cue phrases and building hand-crafted lexicons where the cue phrases are assigned to one of a number of fixed categories. A primary aim of the current research is to investigate whether the effects of T&M's cue phrase features can be achieved using automatically computable linguistic features. If they can, then this helps to relieve the burden involved in porting systems such as these to new domains. Our preliminary cue phrase feature set includes syntactic features of the main verb (voice, tense, aspect, modality, negation), which we have shown to be correlated with rhetorical status [7]. We also use features indicating sentence initial part-of-speech and sentence initial word features to roughly approximate formulaic expressions which are sentence-level adverbial or prepositional phrases. Subject features include the head lemma, entity type, and entity subtype. These features approximate the hand-coded agent features of T&M. A main verb lemma feature simulates T&M's *type of action* and a feature encoding the part-of-speech after the main verb is meant to capture basic subcategorisation information.

4.2 Weka Results and Discussion

We ran experiments with four classifiers in the *Weka* package using default parameter settings: an implementation of Quinlan's [23] decision tree algorithm (C4.5); an implementation of John and Langley's [12] naïve Bayes algorithm incorporating statistical methods for nonparametric density estimation of continuous variables (NB); an implementation of Littlestone's [15] algorithm for mistake-driven learning of a linear separator (Winnow); and an implementation of Platt's [22] sequential minimal optimization algorithm for training a support vector classifier using polynomial kernels (SVM). All accept continuous features as input except Winnow. In order to evaluate the Winnow algorithm, we discretise continuous features using the *Weka* filter based on Fayyad and Irani's [6] MDL method for discretisation.

Micro-averaged F-scores for each classifier are presented in Table 3.[3] The I columns contain individual scores for each feature type and the C columns contain scores which incorporate features incrementally. C4.5 performs very well (65.4) with location features only, but is not able to successfully incorporate other features for improved performance.[4] SVMs perform second best (60.6) with all features. NB is next (51.8) with all but thematic word features. Winnow has the poorest performance with all features giving a micro-averaged F-score of 41.4. For the most part, these scores are considerably lower than the micro-averaged F-score of 72.0 achieved by T&M. However, the picture is slightly different when we con-

[3] Micro-averaging weights categories by their frequency in the corpus. By contrast, macro-averaging puts equal weight on each class regardless of how sparsely populated it might be.

[4] Due to this non-monotonicity and due to the opacity of the highly complex model produced from location features, we consider C4.5 unreliable for our task.

	C4.5		**NB**		**Winnow**		**SVM**	
	I	C	I	C	I	C	I	C
Cue Phrases	47.8	47.8	39.6	39.6	31.1	31.1	52.1	52.1
Location	**65.4**	54.9	34.9	47.5	34.2	40.2	35.9	55.0
Entities	35.5	54.4	32.6	48.8	26.0	40.2	33.1	56.5
Sent. Length	27.2	55.1	20.0	49.1	27.0	40.4	12.0	56.8
Quotations	28.4	59.5	29.7	**51.8**	23.3	41.1	27.8	60.2
Them. Words	30.4	59.7	21.2	51.7	25.7	**41.4**	12.0	60.6
Baseline	12.0							

Table 3: Micro-averaged F-score results for rhetorical classification

sider the systems in the context of their respective baselines. Teufel and Moens [28] report a macro-averaged F-score of 11 for always assigning the most frequent rhetorical class, similar to the simple baseline they use in earlier work. This score is 54 when micro-averaged because of the skewed distribution of rhetorical categories (67% of sentences fall into the most frequent category). With the more uniform distribution of rhetorical categories in the HOLJ corpus, we get baseline numbers of 6.2 (macro-averaged) and 12.0 (micro-averaged). Thus, the actual per-sentence (micro-averaged) F-score improvement is relatively high, with our system achieving an improvement of between 29.4 and 53.4 points (to 41.4 and 65.4 respectively for the optimal Winnow and C4.5 feature sets) where the T&M system achieves an improvement of 18 points. Like T&M, our cue phrase features are the most successful feature subset (excepting C4.5 decision trees). We find these results very encouraging given that we have not invested any time in developing cue phrase features but rather have attempted to simulate these through fully automatic, largely domain-independent linguistic information.

5 Experiments with Maximum Entropy and Sequence Modelling

5.1 Maximum Entropy Classification

Maximum entropy (ME) modelling is another machine learning method which allows the integration of diverse information sources. Though ME approaches are not as good as other machine learning approaches (e.g. vector methods) at modelling the interaction between features, they have proved highly effective in natural language tasks with large, noisy feature sets such as text categorisation, part-of-speech tagging, and named entity recognition. We use a publically available version of an ME estimation toolkit[5] which contains C++ implementations of the LMVM [16] and GIS [5] estimation algorithms.[6] As with Winnow, the *Weka* implementation of Fayyad and Irani's [6] MDL algorithm is used to discretise numeric features. Individual and cumulative feature results are found in the MXT column of Table 4.[7]

Although ME approaches have proved very successful for natural language tasks, they are not in common use in the text summarisation community. Teufel and Moens [28] state simply that they experimented with maximum entropy but it did not show significant improvement over naïve Bayes. We hypothesise that this is due to the very carefully constructed feature set optimised for naïve Bayes. Results from Osborne [21], where maximum entropy was shown

[5]Written by Zhang Le: `http://www.nlplab.cn/zhangle/maxent_toolkit.html`

[6]All final results presented in sections 5.1 and 5.2 use GIS parameter estimation.

[7]Note that while the *Weka* experiments use 10-fold cross-validation, the maximum entropy experiments use per-Lord cross-validation in anticipation of the sequencing experiments where individual Lord's speeches should remain intact.

	MXT		PL		SEQ	
	I	C	I	C	I	C
Cue Phrases	48.1	48.1	51.6	51.6	52.6	52.6
Location	42.5	51.9	38.0	54.0	39.5	56.2
Entities	35.8	53.7	32.0	55.2	35.5	56.5
Sent. Length	21.5	54.0	28.6	56.4	27.9	58.1
Quotations	25.7	57.3	28.5	57.7	30.5	**61.2**
Them. Words	27.7	**57.5**	26.7	**58.1**	31.7	60.8
Baseline	12.0					

Table 4: Maximum entropy F-score results for rhetorical classification.

to perform much better than naïve Bayes when features are highly dependent, support this hypothesis. Our results (Table 4) also support this hypothesis. The feature subset containing the most inter-dependencies in our system is that which uses automatically generated linguistic information to represent cue phrase information. Comparing scores for this feature set, we see that the ME classifier performs nearly 10 points better than naïve Bayes. Maximum entropy outperforms the other classifiers as well for most feature types, falling short only of the C4.5 decision tree on location features and the SVM on cue phrase and quotation features, though the cumulative numbers indicate that it is not integrating diverse information as well as the SVM does. We believe this is due to the SVM being better able to model feature interactions. Explicitly conjoining features in maximum entropy will allow us to test this.

5.2 Sequence Modelling

Order is a general characteristic of natural languages that distinguishes many problems from classification tasks in other domains.[8] For example, when predicting a word's part-of-speech, a classifier should consider the surrounding labels to approximate syntactic constraints. Likewise, it is important in named entity recognition to consider the context of boundary and entity type predictions. Order is also implicit in sentence-level tasks where label contexts capture discourse constraints. The rhetorical status classification task falls in this category since sentences of the same class tend to cluster together in blocks.

There are a number of approaches to sequence modelling in the literature. Hidden Markov models have been the standard for speech applications for some time and have also been applied to word-level tasks such as named entity recognition and shallow parsing, e.g. [30, 20]. Maximum entropy Markov models (MEMMs) and conditional random fields (CRFs) have also been proposed for sequence modelling. In this work, we implement the sequence modelling approach used by Ratnaparkhi [24] for part-of-speech tagging and also used by Curran and Clark [3, 4] for supertagging and named entity recognition. Here, the conditional probability of a tag sequence $y_1..y_n$ given a Lord's speech $s_1..s_n$ is approximated as:

$$p(y_1..y_n|s_1..s_n) \approx \prod_{i=1}^{n} p(y_i|x_i) \tag{1}$$

where $p(y_i|x_i)$ is the normalised probability at sentence i of a tag y_i given the context x_i. $p(y_i|x_i)$ has the following log-linear form:

$$p(y_i|x_i) = \frac{1}{Z(x_i)} exp(\sum_j \lambda_j f_j(x_i, y_i)) \tag{2}$$

[8] The biomedical domain is a notable exception. Order is also implicit in gene sequencing tasks, for instance.

where the f_j include the features described in section 4.1 and features defined in terms of the previous two tags. This framework is very similar to that of MEMMs, a graphical framework that separates transition functions for different source states [17]. However, Ratnaparkhi's model allows arbitrary state-transition structures, and because it combines all of the different source states into a single exponential model, it is likely to cope better with sparse data. Table 4 gives the results for sequencing (SEQ) as well as results for a model incorporating previous labels but no search (PL) and results on the original feature set (MXT). Sequence modelling provides significant improvements over the classifier scores, the optimal configuration achieving an F-score gain of 3.7 points over the optimal classification configuration. Further improvements might be be gained by using a search that incorporates following predictions as well as previous predictions or a re-ranking method, e.g. [2].

6 Conclusions and Future Work

We have presented classifier experiments in the context of summarisation of legal texts, for which we are developing a new corpus of UK House of Lords judgments with detailed linguistic markup in addition to rhetorical status annotation. We have compared a number of machine learning algorithms that have previously shown good performance on natural language tasks. Among these, support vector machines and maximum entropy models prove to be the best suited to our task. We presented a sequence modelling approach to a sentence-level natural language task. This improved performance significantly over the basic classifier.

While generic linguistic analysis tools (e.g. part-of-speech tagging, chunking) are easy to come by in many languages, detailed named entity recognition may not be available for a given new domain. We have invested a considerable amount of time in writing NER rules by hand for the HOLJ domain. Effective bootstrapping methods for NER will make our linguistic features fully domain-independent for domains and languages where linguistic analysis tools are available. In current research, we are exploring the use of active learning to minimise the time and labour needed to create state-of-the-art systems for named entity recognition in novel domains such as astronomy and law.

Finally, on the system level, we are currently developing the sentence extraction component. The core of this component will be a classifier that predicts whether or not a sentence is a good summary sentence. Once this is finished, we will have the building blocks for our summaries. Content will initially be structured using rhetorical templates. We will then be ready to carry out user studies to assess the quality and utility of our system's output and compare our summary text structuring with other methods such as Lapata's [14] probabilistic approach.

References

[1] Nancy A. Chinchor. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Fairfax, Virginia, 1998.

[2] Michael Collins. Discriminative reranking for natural language parsing. In *Proceedings of ICML-2000*, 2000.

[3] James R. Curran and Stephen Clark. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of EACL'03*, 2003.

[4] James R. Curran and Stephen Clark. Language independent NER using a maximum entropy tagger. In *Proceedings of CoNLL-03*, 2003.

[5] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

[6] Usama Fayyad and Keki Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of IJCAI'93*, 1993.

[7] Claire Grover, Ben Hachey, Ian Hughson, and Chris Korycinski. Automatic summarisation of legal documents. In *Proceedings of ICAIL 2003*, Edinburgh, Scotland, 2003.

[8] Claire Grover, Ben Hachey, and Chris Korycinski. Summarising legal texts: Sentential tense and argumentative roles. In *Proceedings of the HLT-NAACL 2003 Text Summarization Workshop*, pages 33–40, Edmonton, Canada, 2003.

[9] Claire Grover, Colin Matheson, Andrei Mikheev, and Marc Moens. LT TTT—a flexible tokenisation tool. In *Proceedings of LREC 2000*, pages 1147–1154, 2000.

[10] Ben Hachey. Recognising clauses using symbolic and machine learning approaches. Master's thesis, University of Edinburgh, 2002.

[11] Ben Hachey and Claire Grover. A rhetorical status classifier for legal text summarisation. In *Proceedings of the ACL-2004 Text Summarization Branches Out Workshop*, 2004.

[12] George H. John and Pat Langley. Esitmating continuous distributions in bayesian classifiers. In *Proceedings of UAI'95*, 1995.

[13] Klaus Krippendorff. *Content analysis: An Introduction to its Methodology*. Sage Publications, Beverly Hills, CA, 1980.

[14] Mirella Lapata. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Meeting of the Association of Computational Linguistics*, pages 545–552, Sapporo, 2003.

[15] Nick Littlestone. Learning quickly when irrelevant attributes are abound: A new linear threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[16] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of CoNLL-2002*, 2002.

[17] Andrew McCallum, Dayne Freitag, and Fernando Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of ICML-2000*, 2000.

[18] Andrei Mikheev. Automatic rule induction for unknown word guessing. *Computational Linguistics*, 23(3):405–423, 1997.

[19] Guido Minnen, John Carroll, and Darren Pearce. Robust, applied morphological generation. In *Proceedings of INLG'2000*, 2000.

[20] Antonio Molina and Ferran Pla. Shallow parsing using specialized HMMs. *The Journal of Machine Learning Research*, 2:595–613, 2002.

[21] Miles Osborne. Using maximum entropy for sentence extraction. In *Proceedings of ACL-2002 Automatic Summarization Workshop*, Philadelphia, USA, July 2002.

[22] John C. Platt. Fast training of support vector machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J.C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1998.

[23] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[24] Adwait Ratnaparkhi. A maximum entropy part-of-speech tagger. In *Proceedings of EMNLP-1996*, 1996.

[25] Erik Tjong Kim Sang and Hervé Déjean. Introduction to the CoNLL-2001 shared task: clause identification. In *Proceedings of CoNLL-2001*, pages 53–57, 2001.

[26] Simone Teufel and Marc Moens. Argumentative classification of extracted sentences as a first step towards fexible abstracting. In *Advances in Automatic Text Summarization*, pages 137–175, New York, 1999. MIT Press.

[27] Simone Teufel and Marc Moens. Discourse-level argumentation in scientific articles: human and automatic annotation. In *Proceedings of ACL-1999 Towards Standards and Tools for Discourse Tagging Workshop*, 1999.

[28] Simone Teufel and Marc Moens. Summarising scientific articles- experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445, 2002.

[29] Henry Thompson, Richard Tobin, David McKelvie, and Chris Brew. LT XML. software API and toolkit for XML processing. `http://www.ltg.ed.ac.uk/software/`, 1997.

[30] GuoDong Zhou and Jian SU. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of ACL-2002*, 2002.