Band 22 – Heft 1 – Jahrgang 2007 – ISSN 0175-1336

Zeitschrift für Computerlinguistik und Sprachtechnologie
GLDV-Journal for Computational Linguistics and Language Technology

LDV Forum

Gesellschaft für linguistische Datenverarbeitung www.gldv.org

LDV-Forum Band 22 Heft 1

Impressum

LDV-Forum **ISSN 0175-1336** **Band 22 – 2007 – Heft 1**	Zeitschrift für Computerlinguistik und Sprachtechnologie GLDV-Journal for Computational Linguistics and Language Technology – Offizielles Organ der GLDV
Herausgeber	Gesellschaft für Linguistische Datenverarbeitung e. V. (GLDV) Juniorprofessor Dr. Alexander Mehler, Universität Bielefeld, *alexander.mehler@uni-bielefeld.de* Prof. Dr. Christian Wolff, Universität Regensburg *christian.wolff@sprachlit.uni-regensburg.de*
Anschrift der Redaktion	Prof. Dr. Christian Wolff, Universität Regensburg Institut für Medien-, Informations- und Kulturwissenschaft D-93040 Regensburg
Wissenschaftlicher Beirat	Vorstand, Beirat und Arbeitskreisleiter der GLDV *http://www.gldv.org/cms/vorstand.php*, *http://www.gldv.org/cms/topics.php*
Erscheinungsweise	2 Hefte im Jahr, halbjährlich zum 31. Mai und 31. Oktober. Preprints und redaktionelle Planungen sind über die Website *http://www.ldv-forum.org* einsehbar.
Einreichung von Beiträgen	Unaufgefordert eingesandte Fachbeiträge werden vor Veröffentlichung von mindestens zwei ReferentInnen begutachtet. Manuskripte sollten deshalb möglichst frühzeitig eingereicht werden und bei Annahme zur Veröffentlichung in jedem Fall elektronisch und zusätzlich auf Papier übermittelt werden. Die namentlich gezeichneten Beiträge geben ausschließlich die Meinung der AutorInnen wieder. Einreichungen sind an die Herausgeber zu übermitteln.
Bezugsbedingungen	Für Mitglieder der GLDV ist der Bezugspreis des LDV-Forums im Jahresbeitrag mit eingeschlossen. Jahresabonnements können zum Preis von 25,- € (inkl. Versand), Einzelexemplare zum Preis von 15,- € (zzgl. Versandkosten) bei der Redaktion bestellt werden.
Satz und Druck	Satz: Carolin Kram, Bielefeld, mit *LaTeX* (*pdfeTeX* / *pdfLaTeX* / *MiKTeX*) und *Adobe InDesign CS2 4.0*, Druck: Druck Team KG, Regensburg

Alexander Mehler und Christian Wolff

Editorial

Liebe GLDV-Mitglieder, liebe Leserinnen und Leser des LDV-Forums,

wir freuen uns, Ihnen die neue Ausgabe des LDV-Forums vorlegen zu können. Diese Ausgabe sticht insofern hervor, als es sich hierbei – nach einer langen Zwischenzeit – um das erste „reguläre“ Heft handelt, das nicht als Themenheft konzipiert ist, sondern vollständig aus einem offenen und international ausgeschriebenen *call for papers* hervorgeht. Damit gelingt im Zuge der Revitalisierung des Forums ein deutlicher Schritt in Richtung des typischen Publikationsmodus einer wissenschaftlichen Fachzeitschrift.

Das vorliegende Heft versammelt unterschiedliche Beiträge aus dem Bereich der Computerlinguistik und angrenzender Disziplinen. Dies betrifft den Bereich der automatischen Diskursanalyse (Caroline Sporleder) und der automatischen Textklassifikation (Edda Leopold, Jörg Kindermann und Gerhard Paaß) sowie den Bereich der Chatbot-Analyse (Bayan Abu Shawar und Eric Atwell auf der einen Seite und Franziskus Geeb auf der anderen Seite). Dass diese vier Beiträge alle im weitesten Sinn mit der Untersuchung von Dialogen zu tun haben, ergibt eine unerwartete, aber hochwillkommene thematische Klammer für das Heft. Abgerundet wird die aktuelle Ausgabe durch einen Beitrag zur Wortsegmentierung im Chinesischen (Xiaofei Lu). Das vorliegende Heft deckt damit nicht nur aktuelle Themenfelder ab, sondern versammelt zugleich eine internationale Autorenschaft. Dass die Mehrheit der Beiträge in englischer Sprache verfasst ist, unterstreicht diesen Trend zur Internationalisierung des Forums.

An dieser Stelle wollen wir Sie erneut auf die zweisprachige Website des LDV-Forums aufmerksam machen, die Sie unter

http://www.ldv-forum.org/

finden. Die Seite informiert Sie über die jeweils aktuelle Ausgabe des Forums und archiviert die Mehrzahl der zurück liegenden Ausgaben, und zwar großenteils im PDF-Format, so dass die Forumsbeiträge nunmehr über Suchmaschinen recherchierbar sind. Beachten Sie bitte, dass wir die Ausgaben des Forums über die digitale Bibliographie dblp, die Sie unter

http://dblp.uni-trier.de/

finden, recherchierbar machen. Dies betrifft insbesondere die BibTeX-Referenzen aller archivierten Forumsartikel. Auf diese Weise ist das Forum in das Spektrum der Zeitschriften- und Konferenzpublikationen aus dem Bereich von Informatik und Computerlinguistik einbezogen und damit einem internationalen Publikum über eine weitere Schnittstelle zugänglich. Die Zweisprachigkeit der Website erhöht hoffentlich die Popularität des Forums und damit langfristig auch der GLDV im gesamten Bereich der Computerlinguistik.

Bitte wenden Sie sich im Falle von Anregungen zur Gestaltung und zum Ausbau der Forums-Website direkt an die Herausgeber. Wir würden uns über Ihre Hinweise und konstruktive Kritik sehr freuen.

Bielefeld und Regensburg im Juni 2007

Alexander Mehler und Christian Wolff

Inhaltsverzeichnis

LDV-FORUM – Band 22(1) – 2007

Alexander Mehler, Christian Wolff
Editorial iii

Inhaltsverzeichnis v

Caroline Sporleder
Manually vs. Automatically Labelled Data in Discourse Relation Classification: Effects of Example and Feature Selection 1

Edda Leopold, Jörg Kindermann, Gerhard Paaß
Analysis of E-Discussions using Classifier Induced Semantic Spaces 21

Bayan Abu Shawar, Eric Atwell
Chatbots: Are they Really Useful? 29

Franziskus Geeb
Chatbots in der praktischen Fachlexikographie und Terminologie 51

Xiaofei Lu
A Hybrid Model for Chinese Word Segmentation 71

Autorenverzeichnis 91

Caroline Sporleder

Manually vs. Automatically Labelled Data in Discourse Relation Classification: Effects of Example and Feature Selection

We explore the task of predicting which discourse relation holds between two text spans in which the relation is not signalled by an unambiguous discourse marker. It has been proposed that automatically labelled data, which can be derived from examples in which a discourse relation is unambiguously signalled, could be used to train a machine learner to perform this task reasonably well. However, more recent results suggest that there are problems with this approach, probably due to the fact that the automatically labelled data has particular properties which are not shared by the data to which the classifier is then applied. We investigate how big this problem really is and whether the unrepresentativeness of the automatically labelled data can be overcome by performing automatic example and feature selection.

1 Introduction

Machine learning approaches have been successfully applied to many areas of natural language processing (NLP). Usually the best results are achieved by *supervised* methods, in which a learner is trained on a set of manually labelled examples. However, manual annotation of training data is time-consuming and costly. In recent years, there has consequently been a shift towards techniques that reduce the annotation effort, either by careful selection of the examples to be annotated (*active learning* (Cohn et al., 1994)) or by mixing labelled and unlabelled data (e.g., via *co-training* (Blum and Mitchell, 1998)). In other cases, heuristics have been used to label some data automatically. One application for which this strategy has been suggested is the identification of discourse relations, such as CONTRAST or EXPLANATION, holding between text spans (Marcu and Echihabi, 2002).

Such discourse relations can be explicitly signalled by discourse connectives. For instance, in example (1) the CONTRAST relation between the two text spans (indicated by square brackets) is signalled by *but*. However, many discourse connectives are ambiguous between several relations, such as *since*, which can signal EXPLANATION (2a) but also a temporal relation (2b). Finally, many relations are not signalled by an explicit discourse marker at all, such as the RESULT relation in (3). Throughout this paper, we will call examples, in which the discourse relation is unambiguously signalled *marked*, and all other examples (i.e., those with an ambiguous marker or no marker at all) will be referred to as *unmarked*. Identifying the correct discourse relation in examples in which the relation is indicated by an unambiguous marker is fairly trivial; one just needs a list of such markers and the relations they map to. If the relation is signalled by an ambiguous marker, the task becomes more difficult and involves disambiguating which of the relations that the marker can signal holds in a given example. But, because the number of distinct relations that an ambiguous marker can signal is typically very limited, the problem should still be relatively easy to solve. If no explicit discourse connective is present, however, the task becomes relatively challenging. In the absence of an explicit marker, a classifier has to rely on other cues, such as the lexical semantics of the words in the spans.

(1) [We can't win,] [**but** we must keep trying.]

(2) a. [I don't believe he's here] [**since** his car isn't parked outside.]
    b. [She has worked in retail] [**since** she moved to Britain.]

(3) [The train hit a car on a level-crossing,] [it derailed.]

Marcu and Echihabi (2002) proposed that the problem could be addressed by utilising the existence of unambiguously marked relations in some examples to extract and automatically label training data for a classifier from a large unannotated corpus. The class label of an extracted example is then assigned on the basis of the unambiguous marker. Example (1), for instance, would be assigned the label CONTRAST. The marker is then removed and a classifier is trained on the automatically labelled data to distinguish between different relations even if no marker is present.

While this approach is very elegant and appealing, more recent studies (Murray et al., 2006; Sporleder and Lascarides, 2007) found evidence that training on automatically labelled data is of limited use for the identification of discourse relations in *unmarked* examples. These results seem to be independent of the classifier used, and it has been suggested (Sporleder and Lascarides, 2007) that the problem stems from the automatically labelled training data themselves, i.e., the fact that the automatically labelled data were originally unambiguously marked means that they are often different from unmarked data which makes it difficult for a classifier trained on the former to generalise to the latter.

In this paper, we aim to explore how big this problem really is. In particular, we explore whether a small set of (unmarked) manually annotated seed data can be exploited to overcome some of the problems associated with automatically labelled data. Such seed data can be used in at least two ways: (i) to select those automatically labelled examples which are most similar to unmarked data (i.e., to the seed data) and therefore, so we hypothesise, make good training examples for the classifier (*example selection*), and (ii) to determine which features generalise best from automatically labelled data to unmarked examples (*feature selection*). We look at both approaches and report the results in sections 4 and 5, respectively. First, however, we give a more detailed overview of the task and previous research (Section 2) and discuss the data and machine learning algorithms that were used in the experiments (Section 3).

2 Discourse Parsing and the Classification of Discourse Relations

Texts are not just random collections of sentences, they have internal structure, which is commonly referred to as *discourse structure*. There are numerous theories of discourse structure, e.g., *Rhetorical Structure Theory* (RST) (Mann and Thompson, 1987), *Discourse Representation Theory* (DRT) (Kamp and Reyle, 1993), *Segmented Discourse Representation Theory* (SDRT) (Asher and Lascarides, 2003) and *Discourse Lexicalised Tree Adjoining Grammar* (DLTAG) (Webber et al., 2003).

Typically, discourse is viewed as a hierarchical structure in which smaller discourse units (also known as *spans*) are connected by *discourse relations* (also known as *rhetorical relations*), such as EXPLANATION, RESULT, or CONTRAST, to form larger units which can then in turn be linked to other discourse units. For example, in (4), the second sentence relates to the first via a RESULT relation and the resulting larger unit links to the third sentence via a CONTINUATION relation (see Figure 1).

(4) a. The high-speed Great Western train hit a car on an unmanned level crossing yesterday.
b. It derailed.
c. Transport Police are investigating the incident.

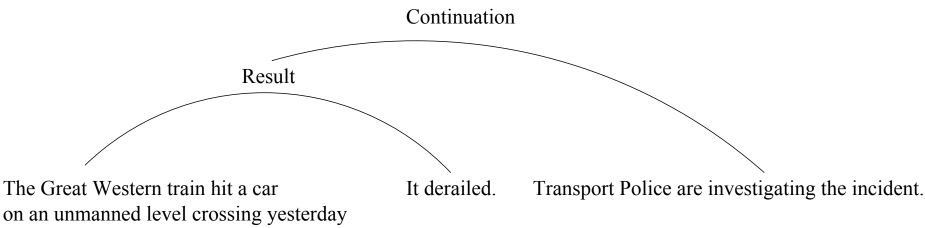


Figure 1: Discourse Structure of Example (4)

Knowledge of the discourse structure of a text would be beneficial for many applications, among them question-answering, information extraction, and text summarisation. As a consequence, there has been a lot of research on *discourse parsing*, i.e., determining the discourse structure of a text by automatic means. Most of the earlier work was rule-based, making use of hand-crafted rules that involved relatively deep semantic analyses (Hobbs et al., 1993; Asher and Lascarides, 2003). A second strand of work comprised rule-based systems that relied more heavily on surface cues than deep semantics (Corston-Oliver, 1998; Polanyi et al., 2004a,b; Le Thanh et al., 2004). With the advent of the first corpora that had been manually annotated with discourse structure, the focus shifted towards systems which employed machine learning techniques to train a discourse parser on these resources (Marcu, 1999; Soricut and Marcu, 2003; Baldridge and Lascarides, 2005).

In this paper, we look at a sub-problem of full-blown discourse parsing, namely identifying the correct discourse relation between two adjacent sentences (*inter-sentential*) or between two clauses within a sentence (*intra-sentential*). We disregard the problem of determining relations in multi-sentence units[1] as well as the problem of determining the correct span boundaries and attachment sites. While these are interesting problems in themselves and, in practice, need to be solved together with the relation classification, the identification of discourse relations in unmarked examples is probably the most challenging sub-task of full discourse parsing. And this sub-task is particularly challenging when it involves the lower levels of a discourse tree, i.e., for relations between sentences or clauses.

Because the identification of discourse relations in unmarked examples is such a complex problem, requiring knowledge of lexical semantics and ideally also some form of world knowledge, machine learning approaches seem to be best suited for solving it. Trained systems achieve reasonably good performance (up to 60% F-score) (Marcu, 1999; Soricut and Marcu, 2003; Baldridge and Lascarides, 2005), but the necessity for annotated corpora is a big limitation. Such corpora are expensive to create and are therefore only available for very few languages. Consequently, there has also been research into how manual annotation of corpora can be avoided or reduced. Nomoto and Matsumoto (1999), for instance, propose an active learning solution for the task of identifying discourse relations between sentences.

Going one step further, Marcu and Echihabi (2002) present an approach which does not require any manual annotation effort at all. Instead they devise a scheme for labelling training data automatically by exploiting the fact that discourse relations are sometimes unambiguously marked. Such examples can be extracted from large unannotated corpora and automatically labelled with the appropriate relation. The discourse markers are then removed and a classifier is trained to identify the correct relation even *in the absence* of an unambiguous marker. This is necessary to adequately process the many examples that either contain no overt discourse marker or that contain a marker that is ambiguous between several relations.

Marcu and Echihabi (2002) applied their method to four relations from Mann and Thompson (1987), namely CONTRAST, CAUSE-EXPLANATION-EVIDENCE, CONDITION and ELABORATION (where CONTRAST and ELABORATION are supertypes for more specific relations in RST). Two types of non-relations (NO-RELATION-SAME-TEXT and NO-RELATION-DIFFERENT-TEXTS) were also included. Marcu and Echihabi identified a number of unambiguous discourse markers for these relations[2] and extracted examples of inter- and

[1]Discourse relations holding between multi-sentence units seem to have a somewhat different distribution than those holding between individual sentences. For example, relations such as TOPIC-SHIFT and SUMMARY are more frequent between multi-sentence segments whereas relations such as EXPLANATION or CONTRAST tend to hold between relatively short spans. Consequently, it has been suggested to implement two different processing strategies for higher and lower level discourse structure (Marcu, 1997).

[2]The non-relations are extracted by randomly choosing pairs of non-adjacent sentences. Because these sentences are non-adjacent it is assumed that no discourse relation holds between them.

intra-sentence relations[3] from a 40 million sentence corpus. They obtained between 900,000 and 4 million examples per relation. The discourse markers were then removed from the extracted data and a Naive Bayes classifier was trained to distinguish between different relations on the basis of co-occurrences between pairs of words, with one word in the pair coming from the left span and the other from the right. Marcu and Echihabi (2002) mainly tested their method on a set of automatically labelled data, i.e., examples which originally contained an unambiguous discourse marker which was used for labelling the example with the gold standard relation and then removed before testing. For this data set they report an accuracy of 49.7% for the six-way classifier. They also tested several binary classifiers (distinguishing between ELABORATION and each of the other relations) on a set of *unmarked* examples. However, they do not report the accuracy or F-score for these experiments, only the recall on the non-ELABORATION relation, which lies between and 44.74% and 69.49%

In later work, Sporleder and Lascarides (2007) investigated more extensively how useful automatically labelled examples are in practice for classifying discourse relations in unmarked examples. They chose five relations from SDRT's inventory of relations (Asher and Lascarides, 2003): CONTRAST, RESULT, EXPLANATION, SUMMARY and CONTINUATION. These relations were selected because for each of them there are discourse markers which signal them unambiguously but they also frequently occur *without* a discourse marker, making it beneficial to be able to determine them automatically if no marker is present. Three corpora were used to extract training data: the British National Corpus (BNC, 100 million words), and two corpora from the news domain — the North American News Text Corpus (350 million words) and the English Gigaword Corpus (1.7 billion words). The number of extracted examples ranged from 8,500 for CONTINUATION to just under 7 million for CONTRAST. In addition, an annotated set of unmarked examples was created by manually labelling data from the RST Discourse Treebank (RST-DT) (Carlson et al., 2002) with SDRT relations. This data set contained 1,051 examples with roughly equal numbers of examples for each of the relations (with the exception of SUMMARY which occurs relatively infrequently in the RST-DT, so only 44 examples were found).

Sporleder and Lascarides (2007) carried out three experiments: (1) training and testing on automatically labelled data (i.e., data in which the relation was originally marked by a discourse marker which was then removed), (2) training on automatically labelled data and testing on unmarked, manually labelled data, and (3) training and testing on manually labelled unmarked data. To investigate whether there are any classifier-specific differences in performance, they employed two different classifiers. The first was a re-implementation of Marcu and Echihabi's (2002) Naive Bayes model and used only lexical features. The other employed a variety of linguistically motivated features (see Sporleder and Lascarides (2007, 2005)) and used the BoosTexter machine learning framework (Schapire and Singer, 2000).

[3]Where the inter-sentence relations involve adjacent sentences. The boundaries of the spans participating in the relation are determined using a set of heuristics based on surface cues.

It was found that training and testing on automatically labelled data led to reasonable results (61% accuracy for the BoosTexter model, 42% for the Naive Bayes model). This suggests that discourse relations are, in principle, learnable from automatically labelled data. However, when the classifiers trained in this way were applied to unmarked examples the performance of both of them dropped to around 26% accuracy. This was just above the baseline of choosing a relation randomly (20% accuracy), though the difference was statistically significant. By comparison, training the BoosTexter model on just 500 manually labelled, unmarked examples led to a noticeably higher accuracy of 40%. A learning curve experiment revealed that just under 140 manually labelled training data were enough to rival the performance that was obtained by training on 72,000 automatically labelled data. Similar findings are reported by Murray et al. (2006) who investigated the learnability of discourse relations in speech.

These results suggest that, (i) it is *possible* to learn discourse relations from automatically labelled data (because training and testing on automatically labelled data led to reasonable results), but (ii) classifiers trained in this way do not generalise well on unmarked data. The fact that both classifiers dropped to similar performance levels indicates furthermore that this is not predominantly a problem with the feature space or the learning framework but that the problem probably stems from the training data itself: Automatically labelled examples, in which the relation was originally marked, may simply be too different from unmarked examples to make good training material for the latter.[4] For instance, a typical marked example of a given relation may exhibit structural properties that are very different from the properties of a typical unmarked example for the same relation. Sporleder and Lascarides (2007) carried out a preliminary study of various linguistic properties of marked and unmarked examples and found some notable disparities, such as a significant variation in span length between marked and unmarked instances of the RESULT relation, and differences in the distribution of part-of-speech tags for the CONTRAST, EXPLANATION, and RESULT relations. In some cases these differences meant that features which were highly predictive of a relation in the marked examples were not predictive of the same relation in the unmarked examples. For instance, CONTINUATION was nearly always holding inter-sententially in the marked examples but occurred more frequently as an intra-sentential relation in the unmarked examples. The predominantly inter-sentential distribution of CONTINUATION in the automatically labelled training data caused the BoosTexter model to learn a decision rule which only predicted CONTINUATION if a relation was holding inter-sententially. While this rule had a relatively high accuracy when applied to the (originally marked) automatically labelled data, it led to a fairly low accuracy on the unmarked data.

[4] An alternative explanation would be that the automatically labelled examples are just too noisy, where the noise comprises mislabelled relations as well as misplaced span boundaries. However, Sporleder and Lascarides (2007) found that their extraction method was fairly accurate, with just 2% of the examples in a manually checked sample containing an error.

3 Data and Machine Learners

In this paper, we investigate whether a small set of manually labelled data can be exploited to automatically select good training data and suitable features which will help to overcome some of the problems associated with training on automatically labelled data. In all experiments we use the same data and the same classifiers that were used by Sporleder and Lascarides (2007).

As mentioned in the previous section, the automatically labelled data was extracted from three corpora: the BNC, the American News Text Corpus and the English Gigaword Corpus. The extracted data covers five relations from SDRT (Asher and Lascarides, 2003): CONTRAST, EXPLANATION, RESULT, SUMMARY, and CONTINUATION. Because the data was highly skewed (with 7 million extracted for CONTRAST but only 8,500 for CONTINUATION) which causes problems for most machine learners, a smaller data set of approximated 72,000 examples was created with roughly uniform distributions across the five relations.

The manually labelled data set contains 1,051 unmarked examples. These were taken from the RST-DT and manually mapped to the corresponding SDRT relations. The distribution of relations in this data set was also approximately uniform, except for SUMMARY of which only 44 unmarked examples were found in the RST-DT.

We used two different classifiers in the experiments. We deliberately chose two classifiers which are fairly different, both with respect to the machine learning technique they use and with respect to their feature space. The first was a re-implementation of the Naive Bayes model proposed by Marcu and Echihabi (2002). This model assumes that the relation that holds between two spans can be determined on the basis of co-occurrences between words.

Let r_i be the discourse relation that holds between two spans W_1 and W_2. The model assumes that r_i can be determined on the basis of the word pairs in the Cartesian product over the words in the two spans: $(w_i, w_j) \in W_1 \times W_2$. The model is derived as follows: Given the assumption in the word pair model, the most likely relation is given by $\underset{r_i}{\operatorname{argmax}}\, P(r_i|W_1 \times W_2)$. According to Bayes rule:

$$P(r_i|W_1 \times W_2) = \frac{P(W_1 \times W_2|r_i)P(r_i)}{P(W_1 \times W_2)} \tag{5}$$

Since for any given example $P(W_1 \times W_2)$ is fixed, the following holds:

$$\underset{r_i}{\operatorname{argmax}}\, P(r_i|W1 \times W_2) = \underset{r_i}{\operatorname{argmax}}\, P(W_1 \times W_2|r_i)P(r_i) \tag{6}$$

We estimate $P(r_i)$ via maximum likelihood on the training set. And to estimate $P(W_1 \times W_1|r_i)$ we assume that all word pairs in the Cartesian product are independent, i.e.:

$$P(W_1 \times W_2|r_i) \approx \prod_{(w_i,w_j)\in W_1 \times W_2} P((w_i, w_j)|r_i) \tag{7}$$

To estimate the probability of a word pair (w_i, w_j) given a relation r_i, we use maximum likelihood estimation and Laplace smoothing. We converted all words in the spans to lower case but — to stay faithful to Marcu and Echihabi (2002) — we did not apply any other pre-processing, such as stemming.

The second model is more complex. It uses a variety of shallow linguistic features (Sporleder and Lascarides, 2005). To combine these features into a classifier we used BoosTexter (Schapire and Singer, 2000), which integrates a boosting algorithm with simple decision rules and allows a variety of feature types, such as nominal, numerical or text-valued features. Text-valued features can, for instance, encode sequences of words or part-of-speech tags. BoosTexter applies n-gram models when forming classification hypotheses for these features (i.e., it tries to detect n-grams in the sequence which are good predictors for a given class label).

We implemented 41 linguistically motivated features, roughly falling into six classes:

- **positional features:** whether the relation holds inter- or intra-sententially, and the position of the example relative to the preceding and following paragraph boundaries
- **length features:** span length
- **lexical features:** words, lemmas, stems, and their overlap, WordNet (Fellbaum, 1998) classes
- **part-of-speech features:** part-of-speech tags
- **temporal features:** finiteness, modality, aspect, voice and negation of the verb phrases
- **cohesion features:** pronoun distribution, presence or absence of ellipses

As some of these features rely on stems, lemmas, and syntactic chunking, we preprocessed the examples with the Porter stemmer (Porter, 1980), the RASP toolkit[5] (Minnen et al., 2001) and the Charniak parser (Charniak, 2000). For a more detailed description of the features and their motivation see Sporleder and Lascarides (2007).

4 Automatic Example Selection

The results reported by Sporleder and Lascarides (2007) suggest that training on automatically labelled data alone does not lead to a satisfactory performance on unmarked examples. This may be because automatically labelled data are derived from examples in which the discourse relation was originally unambiguously marked and these marked examples may be structurally too different from unmarked examples to make good training material for a classifier that is then applied to unmarked data. However, it may be that not all automatically labelled instances are equally bad training material. Marked and unmarked examples are not per se structurally different. Sometimes

[5]Downloadable from `http://www.informatics.susx.ac.uk/research/nlp/rasp/` (20.4.2007).

a discourse marker can be added or removed from a pair of spans without rendering the example infelicitous, as in (8) below. An automatically labelled example (8a) would be indistinguishable from a manually labelled example (8b) and should thus make an equally good training instance. Moreover, even in cases where there is no complete one-to-one correspondence between marked and unmarked examples, the resulting automatically labelled examples are not necessarily useless; as long as they are not too different from unmarked examples a classifier might still be able to learn something from them.

(8) a. She doesn't make bookings **but** she fills notebooks with itinerary recommendations.

b. She doesn't make bookings, she fills notebooks with itinerary recommendations.

Hence, the question is whether it is possible to automatically select those examples that are useful and informative for the classifier. In most automatic example selection approaches, it is assumed that informative examples are those about whose class label the learner is least certain (*uncertainty sampling* (Lewis and Catlett, 1994)). This is the idea behind active learning, where the aim is to reduce the annotation effort by selecting those examples for annotation whose class label cannot yet be predicted confidently, which often means that a smaller training set obtained via active learning leads to models with comparable performance to those that are trained on much larger randomly selected training sets (e.g., Baldridge and Osborne (2004)). In our case, the aim is somewhat different: instead of selecting those examples about which the learner is least certain, we need to select those examples which are most useful for training a classifier that can predict the discourse relations in *unmarked examples*.

One way of doing this would be by using a small manually labelled set of unmarked examples to evaluate the effect of adding and deleting a particular example (or a subset of examples) from the marked, automatically labelled training set. This way the set of automatically labelled examples could be searched for a good subset for training, i.e., a subset which is 'representative' of the unmarked examples that we wish to model. This is called a *wrapper approach* (Blum and Langley, 1997). Wrapper approaches use the induction algorithm itself (in our case the BoosTexter or Naive Bayes model) to determine which examples to include in the final training set. An alternative to the wrapper approach is a *filter approach*, where the training examples are selected independently of the induction algorithm, using some other measure to decide which examples to include.

It has been argued that wrapper methods often lead to better results because they take the bias of the main induction algorithm into account (see Kohavi and John (1997) in the context of feature selection). However, they have the disadvantage of being computationally more expensive than filter methods. Therefore, we opted for a filter approach. We made the simplifying hypothesis that "good" training examples are those which are most similar to manually labelled examples. We could apply a pre-defined similarity criterion, e.g., based on feature vector overlap, to identify those automatically labelled examples which are most similar to the examples in our manually labelled development set. However, we decided to take a different approach and train a classifier to distinguish between automatically and manually labelled examples. We then applied this classifier to the automatically labelled data and selected those examples which the classifier labels as "manual" with the highest probability.

Note that we require a manually labelled data set for our example selection. This means that—from a practical perspective—it is not enough that we can show that training on automatically selected examples leads to a better performance on the manually labelled test set than training on randomly selected examples. Instead, to justify the use of automatically labelled examples, we have to show that using the selected examples leads to a better performance than training on the manually labelled development set alone. That is, we should show that it is possible to bootstrap a classifier by starting with a small set of manually annotated examples and then use these to select further examples from a pool of automatically labelled data and thereby enhance the performance of the original classifier. The baseline for the experiments in this section is therefore the performance of our two models (i.e. the BoosTexter and the Naive Bayes model) when trained on the development set alone.

For comparison, we also investigate what happens if a manually labelled data set is mixed with *randomly* selected automatically labelled data. These experiments are reported in the following section. Section 4.2 then discusses the 'similarity-based' example selection method in more detail, and examines its effects on the performance of the models.

4.1 Random Example Selection

To determine whether the performance of training on a small set of manually labelled examples can be improved upon by adding randomly selected automatically labelled examples, we split the manually labelled set in two halves of 525 instances, one for testing and the other to merge with the automatically labelled examples for training. We then created 15 sets of randomly selected automatically labelled examples of increasing sizes, ranging from 20% of the manually labelled training set (105 automatically labelled instances) to 300% (1,575 instances). The distribution of relations was kept uniform in each sample. We merged each sample with our manually labelled training set, trained the classifiers and then tested on the unseen test set. Then we swapped the manually labelled test and training sets, re-trained and re-tested and averaged the results. Figure 2 shows the learning curves obtained in this way.

For the Naive Bayes word pair model, it can be observed that adding automatically labelled examples generally improves the accuracy. This is due to the fact that this model relies on word features alone and is thus particularly sensitive to sparse data problems. However, big variations in accuracy begin to occur at sampling rates above 100%, where the automatically labelled examples start to dominate the training set. That is, the Naive Bayes model seems to be fairly sensitive to the quality of the training data. Despite this variation, it can be observed that the curve flattens at around 25% accuracy. This level is first achieved for a 160% sampling rate (1,365 training instances overall), and adding further automatically labelled examples does not significantly im-

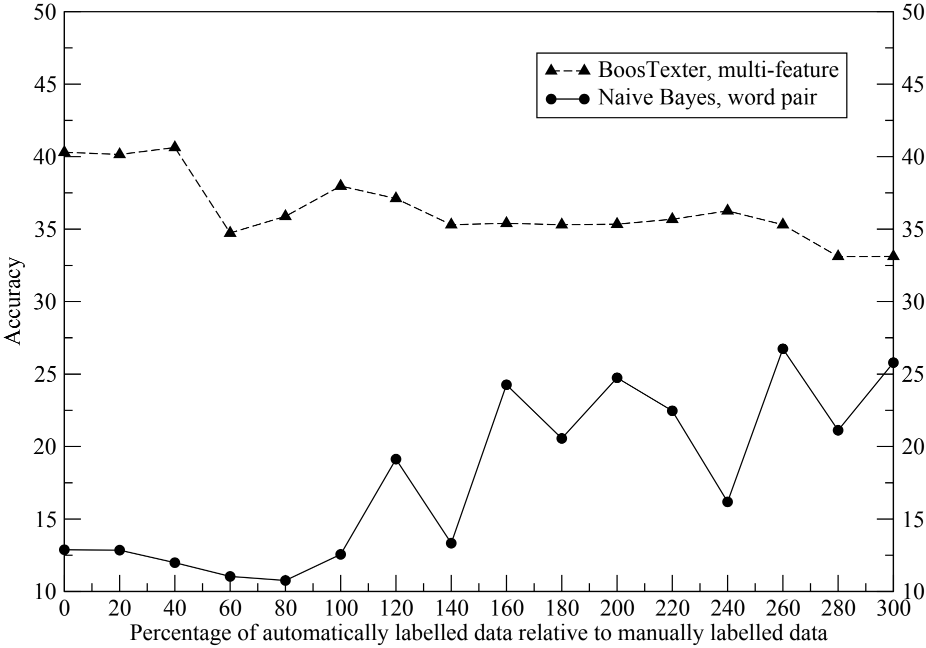


Figure 2: Learning curve for mixing manually and automatically labelled data, averaged over two manually labelled training and test sets

prove the result. Note that training the model on the whole set of automatically labelled examples (around 72,000 instances) and then testing on the manually labelled data also led to an accuracy of around 25%. So it looks like this is the maximum that can be obtained with this model when training on mostly automatically labelled examples and testing on unmarked data.

For the BoosTexter model the situation is different. This model performs relatively well when trained on a small set of manually labelled data and adding randomly selected automatically labelled examples generally decreases the accuracy. At a sampling rate of 300% the accuracy is still around 33% (compared with 40.3% accuracy when trained only on manually labelled data) but, if more automatically labelled examples were added, one would expect this to fall further to around 25%, as this is the level of performance achieved by training on the whole set of automatically extracted examples. Note that there is less variance in the learning curve for the BoosTexter model.

This suggests that this model is better at making the most of each training set.

4.2 Using Machine Learning to Filter Examples

In the previous section, we saw that augmenting an unmarked, manually labelled seed training set with randomly selected, automatically labelled examples hurts performance, at least for the better performing BoosTexter model, which is less sensitive to sparse data. The fact that automatically labelled training examples can hurt performance on the manually labelled test set provides further evidence that the two data sets are linguistically quite different. Given the potential syntactic and semantic differences between them, it makes sense to use machine learning to estimate which of the automatically labelled examples are similar to the manually labelled set, and to use these to augment the seed training set.

To this end, this section presents a more sophisticated example selection method than choosing them randomly. We only ran this experiment for the BoosTexter model as the Naive Bayes word pair model never reached an accuracy of more than 26% in the previous experiments, hence it is unlikely that a more sophisticated sampling method will cause it to outperform the 40% accuracy obtained by the BoosTexter model when trained on the manually labelled data. Moreover, boosting the training set with randomly selected automatically labelled examples actually hurt the performance of the BoosTexter model, and we want to see if other methods of selection can reverse this.

To determine which instances are similar to unmarked examples, we used BoosTexter to train a classifier to distinguish between manually and automatically labelled examples. The training set for this classifier was created by merging half of the manually labelled examples with an equal number of randomly selected automatically labelled examples (with a uniform distribution of rhetorical relations) and replacing the original class labels by a new label encoding whether an instance came from the manually or the automatically labelled set. In a similar way we also created a test set (using the other half of the manually labelled data) so that we could determine how well the classifier could distinguish between the two types of data. We kept the original feature space for our new binary, manual-vs.-automatic classifier.

Table 1 shows the results of applying the binary classifier to the test set. In this table we also report the number of instances for which a given class was predicted (*pred.*), the number of instances labelled with a given class in the gold standard (*GS*), and the number of instances which were correctly labelled by the classifier (*correct*). It can be observed that the accuracy and F-score of the classifier are relatively high at 74.95% and 73.47%, respectively. This is well above the 50% accuracy baseline of choosing a class randomly. Thus it seems that automatically and manually labelled examples can indeed be distinguished to some extent in our feature space. Note, however, that the classifier predicts “manual” more frequently (774 times) than “automatic” (276 times), hence some of the automatically labelled examples seem to be similar enough to manually labelled examples to be assigned to the “manual” class. The question is whether this set of examples can be used to boost the performance of the discourse relation classifier when added to a small set of manually labelled examples.

Class	Avg. Acc	Avg. Prec	Avg. Rec	Avg. F-Score	pred.	GS	correct
manual	n/a	66.93	98.67	79.76	774	525	518
automatic	n/a	97.46	51.24	67.17	276	525	269
all	74.95	82.20	74.96	73.47	n/a	1,050	787

Table 1: Testing the automatic-vs.-manual classifier

To test this we applied the automatic-vs.-manual classifier to the set of automatically extracted training examples. Around 66% of the instances were classified as "manual" by our binary classifier. The confidence of the classifier in the class label is reflected in the weight it assigns to it. When deciding which automatically extracted examples to add to the manually labelled training set, we could choose those which are most confidently predicted to belong to the "manual" class, i.e., those with highest weight for that class. However, we took a slightly different approach: instead of choosing the absolutely highest scoring examples, we selected randomly from the top 10%. The motivation for this is that, while we want to select examples which are similar to manually labelled instances, we also want to select examples which are informative for the learner, i.e., those from which something new can be learnt. Randomly sampling from the top 10% of examples ensures that we select examples which are similar to the manually labelled instances but we do not necessarily select the *most* similar ones.

Using this strategy we selected 525 automatically labelled examples, making sure that the distribution of rhetorical relations was uniform. We merged this set with the 525 manually labelled examples that we had trained our binary, automatic-vs.-manual classifier on. The resulting set was then used as training material for the relation classifier and the trained model was tested on the other half of the manually labelled data (which was not used in the example selection process).

Table 2, which is taken from Sporleder and Lascarides (2007), shows the result of training on 535 manually labelled examples alone (averaged over two runs). This is the baseline that we would hope to beat by adding carefully selected automatically labelled training data to the manually labelled seed data set. Table 3 shows the results of training on 525 manually labelled data and an equal amount of automatically selected automatically labelled data. It can be observed that our sampling strategy does not lead to an improved performance over training on the manually labelled data alone; on the contrary the accuracy drops by around 5%. For comparison, Table 4 shows the results of randomly selecting 525 automatically labelled examples (averaged over five random samples). Random sampling actually leads to a slightly better performance than our machine learning based sampling strategy, though this difference is not significant ($\chi^2 = 1.54, DoF = 1, p <= 0.22$). There could be several reasons for this. First, it is

Relation	Avg. Acc	Avg. Prec	Avg. Rec	Avg. F-Score
CONTINUATION	n/a	36.78	36.85	36.77
RESULT	n/a	38.53	46.32	41.99
SUMMARY	n/a	13.75	3.64	5.63
EXPLANATION	n/a	49.80	50.15	49.85
CONTRAST	n/a	36.70	32.21	34.19
all	40.30	35.11	33.83	33.69

Table 2: Baseline: Training and Testing on Manually Labelled Data, 5 times 2-fold cross-validation averaged

possible that our underlying assumption that automatically labelled examples which are similar to manually labelled ones make good training material is wrong. Second, it could be that this hypothesis is valid but that the problem lies with our automatic-vs.-manual classifier, i.e., it may be that this classifier is simply not accurate enough (though we have found it to achieve an accuracy of above 75%).

Relation	Avg. Acc	Avg. Prec	Avg. Rec	Avg. F-Score
CONTINUATION	n/a	38.06	39.23	38.64
RESULT	n/a	29.81	23.31	26.16
SUMMARY	n/a	7.14	4.55	5.56
EXPLANATION	n/a	40.72	50.75	45.19
CONTRAST	n/a	32.50	31.94	32.22
all	35.24	29.64	29.95	29.55

Table 3: Training on 50% manually labelled and 50% automatically selected automatically labelled examples

5 Using Manually Labelled Data for Automatic Feature Selection

In the previous section, we experimented with automatic example selection and found that adding training examples selected in this way to a manually annotated seed set does not lead to any improvements compared to training on the manually labelled data alone. On the contrary, adding automatically labelled examples decreases the accuracy of the classifier and it does not seem to matter whether the examples are selected randomly or using the similarity-based approach. In this section, we explore the effect of automatic feature selection. Sporleder and Lascarides (2007) argued that one reason

Relation	Avg. Acc	Avg. Prec	Avg. Rec	Avg. F-Score
CONTINUATION	n/a	38.27	38.93	38.54
RESULT	n/a	37.06	37.07	37.06
SUMMARY	n/a	12.67	18.18	14.84
EXPLANATION	n/a	43.62	46.27	44.77
CONTRAST	n/a	39.57	31.55	34.64
all	37.97	34.24	34.40	33.94

Table 4: Training on 50% manually labelled and 50% randomly selected automatically labelled examples, averaged over 5 sampling runs

why a classifier that was trained on automatically labelled data did not generalise well to unmarked examples, could be that —due to structural differences between the two types of data— some features that are predictive for a given relation on one type of data are not predictive on the other type of data and vice versa. One way to overcome this problem could be by automatically selecting those features which maximise the classifier's performance on unmarked data. This can be done by testing individual features and feature combinations on a small manually annotated seed data set (the *development* set) and selecting only the best performing ones.

The feature selection was only performed for the BoosTexter model, since that was found to consistently lead to better results than the Naive Bayes model. We employed a greedy, wrapper-based feature selection strategy. The manually labelled data was split into two parts with equal proportions of the five relations: 20% (i.e., 208 examples) were used as a development set in the feature selection process, the remaining 80% (843 examples) were used for testing. The selection process was started off by training a one-feature classifier for each of the 41 features in our complete feature set. The classifiers were trained on the complete set of automatically labelled examples and then tested on the 208 manually labelled examples in the development set. The best performing feature was then selected. In the next round each of the remaining 40 features was added individually to the best performing feature from the previous round. The resulting 40 2-feature classifiers were trained and tested again, and the best performing 2-feature set was used as the basis to which new features were added in the next round and so on. The feature selection process stopped when adding features did not lead to any improvements in accuracy on the development set. Once the features had been selected, a classifier was trained on the automatically labelled data using only the selected features. The classifier was then tested on the remaining 80% of the manually labelled test set. To abstract away from possible idiosyncracies of the test and development set, we ran the experiment five times, each time with a different 80:20 split of test and development data. The results of the five experiments (*Run 1* to *Run 5*) are reported in Table 5. The table lists the number of features selected (# *Features*), the accuracy that was achieved with the selected features on the development set (*Devel. Acc.*) and on the unseen test set (*Test Acc.*), and the test set accuracy that was obtained with the whole feature set (*Test Acc. all feat.*). From the results it is evident that feature selection does not have a noticeable positive effect on the performance of the classifier. While the test set accuracy with the reduced feature set is often somewhat higher than with the full set (e.g., for Run 1), this difference was not found to be significant in any of the cases. In those cases where training on the full feature set leads to better results, the difference was also not significant.

	# Features	Devel. Acc.	Test Acc.	Test Acc. all feat.
Run 1	3	30.29%	24.08%	24.04%
Run 2	5	35.10%	24.32%	25.27%
Run 3	3	29.33%	25.86%	25.86%
Run 4	4	33.17%	27.16%	25.86%
Run 5	3	31.51%	26.68%	25.84%
Avg.	3.6	31.88%	25.62%	25.37%

Table 5: Feature Selection on a Manually Labelled Development Set

Table 5 also shows that the feature selection algorithm overfits on the development set, i.e., the performance on the development set is noticeably higher than on the test set. This might be due to the somewhat simplistic selection algorithm, which only stops adding features when *no* accuracy gain can be obtained anymore. Having a more sophisticated stopping criterion would probably reduce this overfitting effect. However, the number of features selected by our algorithm is generally very small (3.6 on average), so stopping earlier would result in only one or two selected features. This is unlikely to lead to any significant improvements as the selected features vary a lot between different runs, (i.e., they depend very much on the data sets that are used), which becomes evident from Table 6 in which the features that were selected in the five runs are listed.

Run 1	dist. prev. paragraph, tense info left, adjective lemma overlap
Run 2	content word lemmas left, verb lemma overlap, dist. prev. paragraph, noun lemmas left, pronouns 2P right
Run 3	words left, span length right, ellipsis left
Run 4	words left, word overlap, noun WordNet classes overlap, noun lemma overlap
Run 5	content word lemma overlap, pronouns 3P left, pronouns 2P right

Table 6: Features Selected on Different Runs (in order of selection)

There is relatively little overlap between the different runs with respect to the selected features;[6] no feature was selected in every runs and only three features were selected in more than one run: the distance from the preceding paragraph boundary (*dist. prev. paragraph*, two runs), the number of second person pronouns in the right span (*pronouns 2P right*, two runs), and the words in the left span (*words left*, two runs). To some extend this variation can be explained by the fact that the lexical features are often not very different from each other (e.g., *words left* vs. *content word lemmas left*), so it may be a matter of coincidence which one is chosen first and once this feature has been chosen it makes no sense to add the other one anymore. However, there is also much variation between different *types* of features. For example, in Run 5, predominantly cohesion features are chosen (*pronouns 2P right*, *pronouns 3P left*) which do not occur much in the other runs. One pattern that does arise, however, is that lexical features generally seem to be quite important, hence all runs include at least one lexical feature. Also, information about the left span seems to be more important than information about the right span. This observation was also reported by Sporleder and Lascarides (2005) who explored which features performed best on an *automatically labelled* test set (hence this is not a difference between manually and automatically labelled data). From a human discourse processing perspective it seems plausible that the left span should contain more information about the upcoming discourse relation: it makes the information easier to process than if the signalling is delayed until the right span.

It is also interesting to note that the features which were identified as potentially problematic by Sporleder and Lascarides (2007) because they encode properties on which marked and unmarked examples tend to differ, tend *not* to be selected. Such features are, for example, those that encode span length, part-of-speech tags, or whether the relation holds inter- or intra-sententially.[7] We performed a number of control experiments in which we ran the feature selection on an automatically labelled development set, and found that those features do get selected in that case. In other words, it looks like the feature selection process is able to identify and avoid the most problematic features, i.e., those which do not generalise from marked to unmarked data. However, this does not seem to be enough to reliably and significantly boost performance.

6 Conclusion

Recent research by Sporleder and Lascarides (2007) and Murray et al. (2006) found evidence that Marcu and Echihabi's (2002) suggestion of using automatically labelled data to train a classifier to determine discourse relations in unmarked examples does not work very well in practice. The likely reason for this is that the two types of examples are too dissimilar, i.e., automatically labelled examples, in which the relation was orig-

[6] To check in how far this variation is due to the size of the development set, we ran a similar experiment with a 50:50 split of the manually labelled data into test and development set (i.e., with 525 examples in each of the sets). For these data set, the algorithm generally only selected on or two features and there was still a fair amount of variation.

[7] The only "problematic" feature that gets selected is *span length right* which was selected in Run 3.

inally marked by an unambiguous discourse connective, are simply not representative of the unmarked examples to which the classifier is applied. In this paper, we investigated how fundamental this problem really is and whether a small set of manually labelled seed data (of unmarked examples) might be harnessed to overcome the unrepresentativeness of automatically labelled examples. In particular, we looked at whether such seed data could be used (i) to select automatically labelled examples which are similar to the unmarked data we wish to model and hence make good training material for the classifier, and (ii) to select features which generalise well from the marked, automatically labelled data to the unmarked data.

We found problems with both approaches. For a classifier which is very sensitive to sparse data, like the word-based Naive Bayes model proposed by Marcu and Echihabi (2002), boosting a small manually labelled seed data set with automatically labelled examples helps to improve performance. But, for our re-implementation of this model, we found that the accuracy does not seem to improve much beyond 26% for a 5-way classification task, no matter how much automatically labelled training data is added. This is hardly an acceptable performance level for any real-world application.[8] For models which are less afflicted by sparse data problems, like our multi-feature BoosTexter model, adding automatically labelled data to a manually labelled seed set actually harms performance, and this effect is already noticeably for relatively small amounts of added data. Moreover, it does not seem to make much difference whether the automatically labelled examples are selected randomly or via a more sophisticated, similarity-based selection strategy.

The feature selection experiments lead to similarly sobering results. While training on a carefully selected reduced feature set often led to somewhat better results than training on the full set, this difference was never significant. Moreover, the feature selection was found to be fairly dependent on the data sets that were used. One positive aspect was that features modelling linguistic properties that Sporleder and Lascarides (2007) identified as varying a lot between marked and unmarked data (e.g., span length, inter- vs. intra-sentential relations, part-of-speech tags) tended not to be chosen. The selected features were mainly lexical, thus it seems that lexical features generalise best from the marked to the unmarked case.

Given these results, it is not clear that automatically labelled data can be turned into a valuable resource for this task. It may be possible that sophisticated lexical features can be developed that do in fact generalise from automatically labelled to unmarked data. But a better strategy would probably be to invest resources in the creation of manually annotated data, e.g., corpora annotated with discourse such as the RST-DT,[9] the *Penn Discourse Treebank*,[10] or the *Potsdam Commentary Corpus*,[11] and in the development of

[8] It should be noted that training on automatically labelled data does not always lead to unacceptable results. While machine learning systems that are trained on such data generally perform less well than those trained on manually labelled data, adding automatically labelled instances to a small manually labelled set can sometimes boost performance, as in co-training (Blum and Mitchell, 1998).

[9] `http://www.ldc.upenn.edu/Catalog/LDC2002T07.html`.

[10] `http://www.seas.upenn.edu/~pdtb/`.

[11] `http://www.ling.uni-potsdam.de/cl/cl/res/forsch_pcc.html`.

good classifiers which can make the most of even a small amount of training data.

Acknowledgements

Part of this work was carried out at the University of Edinburgh, funded by EPSRC grant number GR/R40036/01. I am grateful to Alex Lascarides for many interesting discussions on this topic.

References

Asher, N. and Lascarides, A. (2003). *Logics of Conversation*. Cambridge University Press.

Baldridge, J. and Lascarides, A. (2005). Probabilistic head-driven parsing for discourse structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*.

Baldridge, J. and Osborne, M. (2004). Active learning and the total cost of annotation. In *Proceedings of Empirical Approaches to Natural Language Processing (EMNLP)*.

Blum, A. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory (COLT)*.

Carlson, L., Marcu, D., and Okurowski, M. E. (2002). RST Discourse Treebank. Linguistic Data Consortium.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the Assocation for Computational Linguistics*, pages 132–139, Seattle, WA.

Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

Corston-Oliver, S. H. (1998). Identifying the linguistic correlates of rhetorical relations. In *Proceedings of the ACL Workshop on Discourse Relations and Discourse Markers*, pages 8–14.

Fellbaum, C., editor (1998). *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

Hobbs, J. R., Stickel, M., Appelt, D., and Martin, P. (1993). Interpretation as abduction. *Artificial Intelligence*, 63(1–2):69–142.

Kamp, H. and Reyle, U. (1993). *From Discourse To Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence Journal*, 1–2:273–324.

Le Thanh, H., Abeysinghe, G., and Huyck, C. (2004). Generation discourse structures for written text. In *Proceedings of COLING-04*, pages 329–335.

Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 148–156.

Mann, W. C. and Thompson, S. A. (1987). Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, ISI, Los Angeles, CA.

Marcu, D. (1997). *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. PhD thesis, Department of Computer Science, University of Toronto.

Marcu, D. (1999). A decision-based approach to rhetorical parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pages 365–372.

Marcu, D. and Echihabi, A. (2002). An unsupervised approach to recognizing discourse relations. In *Proceedings of ACL-02*, pages 368–375.

Minnen, G., Carroll, J., and Pearce, D. (2001). Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Murray, G., Taboada, M., and Renals, S. (2006). Prosodic correlates of rhetorical relations. In *Proceedings of the HLT-NAACL ACTS Workshop*.

Nomoto, T. and Matsumoto, Y. (1999). Learning discourse relations with active data selection. In *Proceedings of EMNLP-99*.

Polanyi, L., Culy, C., van den Berg, M., Thione, G. L., and Ahn, D. (2004a). A rule based approach to discourse parsing. In *Proceedings of the 5th SIGDIAL Workshop in Discourse and Dialogue*, pages 108–117.

Polanyi, L., Culy, C., van den Berg, M., Thione, G. L., and Ahn, D. (2004b). Sentential structure and discourse parsing. In *Proceedings of the ACL-04 Workshop on Discourse Annotation*.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14:130–137.

Schapire, R. E. and Singer, Y. (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Soricut, R. and Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Sporleder, C. and Lascarides, A. (2005). Exploiting linguistic cues to classify rhetorical relations. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-05)*.

Sporleder, C. and Lascarides, A. (2007). Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*. to appear.

Webber, B. L., Knott, A., Stone, M., and Joshi, A. (2003). Anaphora and discourse structure. *Computational Linguistics*, 29(4):545–588.

Edda Leopold, Jörg Kindermann, Gerhard Paaß

Analysis of E-Discussions Using Classifier Induced Semantic Spaces

We categorise contributions to an e-discussion platform using Classifier Induced Semantic Spaces and Self-Organising Maps. Analysing the contributions delivers insight into the nature of the communication process, makes it more comprehensible and renders the resulting decisions more transparent. Additionally, it can serve as a basis to monitor how the structure of the communication evolves over time. We evaluate our approach on a public e-discussion about an urban planning project, the Berlin Alexanderplatz, Germany. The proposed technique does not only produce high-level-features relevant to structure and monitor computer mediated communication, but also provides insight into how typical a particular document is for a specific category.

1 Introduction

E-discussion platforms facilitate the moderated collaboration between persons distributed in time and space. In order to support a focussed and goal-oriented discussion it is desirable to provide condensed information about the ongoing discussion process in order to monitor and to influence the discourse.

We propose to analyse e-discussion contributions on the basis of Classifier Induced Semantic Spaces (CISSs) (Leopold et al., 2004; Leopold, 2005), and visualise the resulting semantic spaces with Self-Organising Maps (SOMs)(Kohonen, 1995). CISSs can be constructed from any supervised classifier, that determines the membership of a given entity to a pre-defined category on the basis of some numerical threshold. Here, we use a Support Vector Machine (SVM) (Vapnik, 1998) as classifier.

A semantic space is a metric space whose elements are representations of signs of a semiotic system. The metric of the space quantifies some semantic dissimilarity of the signs. If the semantic space is a vector space then its dimensions are associated with some kind of meaning.

Self-Organising Maps are a technique to map elements of a vector-space with three or more dimensions into a two-dimensional "map" by preserving the original distance relationships as far as possible. They therefore allow to represent the structure of a semantic space in the two dimensions of a computer screen.

The remainder of the paper is organised as follows. In Section 2, we discuss the integral parts of the proposed method of structuring the components of a communication network built from e-discussion contributions. In Section 4, we describe the experimental evaluation of our approach, and in Section 5, we conclude.

2 Discourse Grammar and E-Discourse

According to Turoff et al. (1999) a discourse is a deliberative, reasoned communication, which is focused and intended to culminate in decision making. Turoff et al. (1999) argued that building a discourse grammar, which allows individuals to classify their contributions according to their pragmatic function within the discourse is a collaborative effort and is an integral part of the discussion process.

Inspired by the Bülerian Organon-Model (Bühler, 1934), we decided to consider a discourse grammar, that consists the following pragmatic functions: 'giving information', 'making an objection', 'asking a question', and 'giving a reply' . So the contributions of the e-discourse can be assigned to four different classes $z_k, k = 1 \ldots 4$, corresponding to the linguistic functions they fulfil. Multiple class assignment is supported.

The assignment to the different classes is performed inductively based on the judgement of human experts. So in contrast to a rule-based approach we avoid to explicitly construct rules that define the pragmatic functions. We think that this makes our approach more flexible as the language-system changes. It may be, however, interesting to combine both inductive learning and deductive construction of rules.

2.1 Classifier Induced Semantic Spaces

A *classifier induced semantic space* (CISS) is generated in two steps: In the training phase classification rules $\vec{x}_j \rightarrow z_k$ are inferred from the training data. In the classification phase these decision rules are applied to possibly not-annotated documents.

Any supervised classifier, that internally calculates a quantity and bases its classification on whether this quantity exceeds a given threshold or not, can be employed to construct a CISS. So Linear Classifier, Naive Bayes classifiers as well as Support Vector Machines are applicable for the construction of as classifier induced semantic space. We decided to use Support Vector Machines (SVM)s because of its efficiency and its ability to handle high-dimensional input spaces.

An SVM is a supervised binary classifier, that takes as input a set $E = \{(\vec{x}_1, y_1), \cdots, (\vec{x}_n, y_n)\}$ of positive and negative training examples, where each entity x_i belongs to an instance space X, and each y_i belongs to a set $Y = \{-1, +1\}$ of binary class labels. In the classification phase an SVM generates a numerical value $v(\vec{x})$ for each instance $\vec{x} \in X$. The instance $\vec{x}$ is considered to belong to the positive class $y = +1$ if $v(\vec{x})$ is above a certain threshold, and to belong to the negative class $y = -1$ otherwise.

In order to construct a CISS, the SVM-classifier is used as follows: *In the training phase* for each contribution of the e-discussion a word-frequency vector $\vec{x}$ is computed. Since the SVM is a binary classifier, one separate SVM is trained for each of the K class-labels. To this end we consider all contributions belonging to the category in question as positive examples ($y = +1$), whereas all others are considered as negative examples ($y = -1$). This approach is usually referred to as a 1 vs. $K - 1$ setting.

In the *classification* phase, each of the K SVMs assigns a value $v_k(\vec{x}), k = 1 \ldots K$ to the contribution $\vec{x}$. A document with word-frequency vector $\vec{x}$ is represented by a vector $\vec{v}(\vec{x}) = (v_1(\vec{x}), \ldots, v_K(\vec{x}))^T$ and the k-th component $v_k(\vec{x})$ can be interpreted as to which degree the instance $\vec{x}$ belongs to class z_k, which in our context means how much contribution $\vec{x}$ fullfils the linguistic function z_k.

This construction of a semantic space is especially useful for practical applications because (1) the space is low-dimensional (up to dozens of dimensions) and thus can easily be visualised, (2) the space's dimension possesses a well defined semantic interpretation, and (3) the space can be tailored to the special requirements of a specific application. (4) Concurrent techniques like latent semantic analysis (LSA, Landauer and Dumais, 1997), probabilistic latent semantic analysis (PLSA, Hofmann, 2001) and hierarchical latent semantic analysis (Paaß et al., 2004) base their notion of semantic nearness on features of the texts i.e. co-occurrences of words, whereas semantic nearness in a CISS is based on the judgement of a human classifier. (5) It is in principle possible to represent units of different semiotic systems (e.g. different languages or even texts and pictures) in one and the same CISS, given that a sufficiently large training set is available.

2.2 Visualisation of Semantic Spaces with Self-Organizing Maps

Self-Organising Maps (SOM) were invented in the early 80s by Kohonen (1980). They use a specific neural network architecture to perform a recursive regression leading to a reduction of the dimension of the data. For practical applications SOMs can be considered as a distance preserving mapping from a more than three-dimensional space to two-dimensions. A description of the SOM algorithm and a thorough discussion of the topic is given in (Kohonen, 1995). After having run the SVM-classification, both labelled and unlabelled contributions were used to build the Self-Organising Map from the the semantic space.

3 The Data

We evaluate our approach on a public e-discussion about an urban planning project, the Berlin Alexanderplatz, Germany. The Berlin Senate office commissioned an Internet-based civic participation in the course of planning the restructuring of one of the great city squares, the Alexanderplatz. An Internet-based discussion bulletin board was established, where citizens could express and discuss their suggestions and preferences with regard to the future shape of the square. The results of the e-discussion have in the meantime been taken into consideration by the city planners. The e-discussion was supervised by several project collaborators acting as moderators. The participants could post messages referring to a list of topics as well as reply to other participant's messages. The moderators used the same means of communication. (Roeder et al., 2005)

All contributions of the participants and moderators were recorded, yielding 1021 messages in total. 216 contributions (21%) have been annotated according to their pragmatic functions described in section 2. The remaining contributions were left unlabelled.

Table 1: Accuracy of the trained classifiers.

function	precision	recall	F-score
information	72.1	67.4	69.7
objection	56.0	76.1	64.6
question	76.9	61.2	68.2
reply	53.5	84.7	65.6

4 Experimental Results

Training on the annotated data results in four SVMs, which are each trained to separate contributions belonging to one linguistic function (positive class) from all other contributions, which together constitute the negative class. The performance of the classifiers was tested prior to the construction of the semantic space. The results are displayed in table 1. By F-score we refer to the harmonic mean of precision and recall, i.e. $F = 2\left(\frac{1}{prec} + \frac{1}{rec}\right)^{-1}$.

The classification performance is significantly above chance ($F \approx 25\%$). Therefore the classifiers are reliable enough for the construction of a semantic space. The joint classification by the four SVMs produces a 4-dimensional classifier induced semantic space. The four dimensions of this space can be associated with the four pragmatic functions described above.

The 1021 contributions to the e-discussion (both labelled and unlabelled data) were represented in the CISS. These data were used to build the Self-Organising Map as a two-dimensional representation of the semantic space admitting a minimum distortion of the original four-dimensional distances between the data points.

Figure 4 shows an example of a SOM visualising the relations of the contributions in terms of their linguistic functions. SVMs for the four linguistic functions 'question', 'reply', 'information', and 'objections' were trained on 216 contributions (21% of the total contributions) that have been annotated according to their linguistic function. Classification and generation of the SOM was performed for the entire discourse of 1021 contributions.

The contributions of one participant of the e-discussion are displayed by white crosses. The categories are indicated by different grey tones. The SOM algorithm is applied (with 70×70 nodes using Euclidean metric) in order to map the four-dimensional document representations to two dimensions admitting a minimum distortion of the distances. The grey tone indicates the topic category. Shadings within the categories indicate the confidence of the estimated class membership.

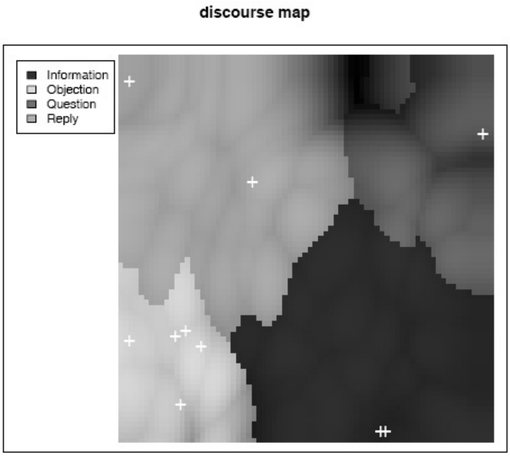


Figure 1: A discourse map generated from a CISS.

We observed that the distribution of authors in the e-discourse does not differ from what is usually observed in the bibliometrical science, namely that the number of authors who publish a given number of papers obeys a power-law. This fact is well known as Lotka's Law (Lotka, 1926). Table 2 shows the frequency distribution that we observed in the Alexanderplatz discussion.

Table 2: Frequency distribution of authors' contributions in the Alexanderplatz e-discussion

♯ contributions	♯ authors	♯contributions	♯ authors
1	73	13	1
2	22	15	1
3	18	24	3
4	4	25	1
5	5	26	1
6	3	29	1
7	3	35	1
9	2	56	1
10	1	140	1
11	2	273	1
12	3		

5 Conclusion

LSA, PLSA, and CISS map documents to the semantic space in a different manner. In the case of LSA the representation of the document in the semantic space is achieved by matrix multiplication. The dimensions of the semantic space correspond to the K largest eigen-values of the covariance matrix. PLSA maps a document to the vector of the conditional probabilities, which indicate how probable aspect z_k is, when a given document is selected. The probabilities are derived from the aspect model using the maximum likelihood principle and the assumption of multinomially distributed word frequency distributions.

The advantage of the presented technique:

1) The use of a supervised classifier makes it possible to produce high-level-features that are relevant to the problem in question (in this case to monitor the discussion process).

2) Note that classifier induced semantic spaces go beyond a mere extrapolation of the annotations found in the training corpus. It gives an insight into how typical a certain document is for each of the classes. Furthermore CISS allow to reveal unseen previously relationships between classes.

3) Concurrent techniques like latent semantic analysis (LSA) (Landauer and Dumais, 1997), probabilistic latent semantic analysis (PLSA) (Hofmann, 2001), and its hierarchical extension (Paaß et al., 2004) base their notion of semantic nearness on features of the texts i.e. co-occurrences of words, whereas semantic nearness in a CISS is based on the judgement of a human classifier.

References

Bühler, K. (1934). *Sprachtheorie*. G. Fischer, Jena.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(6):177–196.

Kohonen, T. (1980). *Content-adressable Memories*. Springer.

Kohonen, T. (1995). *Self-organising Maps*. Springer.

Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory. *Psychological Review*, 104(2):211–240.

Leopold, E. (2005). On semantic spaces. *LDV-Forum*, 18(3):63–86.

Leopold, E., May, M., and Paaß, G. (2004). Data mining and text mining for science & technology research. In Moed, H. F., Glänzel, W., and Schmoch, U., editors, *Handbook of Quantitative Science and Technology Research*, pages 187–214. Kluwer.

Lotka, A. (1926). The frequency distribution of scientific productivity. *Journal of the Washington Academy of Science*, 16(12):317–323.

Paaß, G., Kindermann, J., and Leopold, E. (2004). Learning prototype ontologies by hierarchical latent semantic analysis. In *Workshop on Knowledge Discovery and Ontologies at the joint European Conferences on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2004)*, September 20–24, Pisa (Italy).

Roeder, S., Poppenborg, A., Michaelis, S., Märker, O., and Salz, S. (2005). Public budget dialogue - an innovative approach to e-participation. In Böhlen, M., Gamper, J., Polasek, W., and Wimmer, M., editors, *Proceedings of the International Conference TCGOV 2005, Bolzano (Italy) March 2–4, 2005*, pages 48–56. Springer Lecture Notes in Computer Science, Number 3416.

Turoff, M., Hiltz, S. R., Bieber, M., Fjemestadt, M., and Ajaz, R. (1999). Collaborative discourse structures in computer-mediated group communications. *Journal of Computer-Mediated Communication*, 4:104–125.

Vapnik, V. N. (1998). *An Introduction to Computational Learning Theory*. Wiley & Sons.

Bayan Abu Shawar, Eric Atwell

Chatbots: Are they Really Useful?

Chatbots are computer programs that interact with users using natural languages. This technology started in the 1960's; the aim was to see if chatbot systems could fool users that they were real humans. However, chatbot systems are not only built to mimic human conversation, and entertain users. In this paper, we investigate other applications where chatbots could be useful such as education, information retrival, business, and e-commerce. A range of chatbots with useful applications, including several based on the ALICE/AIML architecture, are presented in this paper.

Chatbots sind Computerprogramme, die mit Benutzern in natürlicher Sprache kommunizieren. Die ersten Programme gab es in den 60er Jahren; das Ziel war festzustellen, ob Chatbots Benutzer davon überzeugen könnten, dass sie in Wirklichkeit Menschen seien. Chatbots werden aber nicht nur gebaut, um menschliche Kommunikation nachzuahmen und um Benutzer zu unterhalten. In diesem Artikel untersuchen wir andere Anwendungen für Chatbots, zum Beispiel in Bildung, Suchmaschinen, kommerzielle Anwendungen und e-commerce. Wir stellen eine Reihe von Chatbots mit nützlichen Anwendungen vor, einschliesslich mehrerer Chatbots, die auf der ALICE/AIML Architektur basieren.

1 Introduction

> The need of conversational agents has become acute with the widespread use of personal machines with the wish to communicate and the desire of their makers to provide natural language interfaces (Wilks, 1999)

Just as people use language for human communication, people want to use their language to communicate with computers. Zadrozny et al. (2000) agreed that the best way to facilitate Human Computer Interaction (HCI) is by allowing users "to express their interest, wishes, or queries directly and naturally, by speaking, typing, and pointing".

This was the driver behind the development of chatbots. A chatbot system is a software program that interacts with users using natural language. Different terms have been used for a chatbot such as: machine conversation system, virtual agent, dialogue system, and chatterbot. The purpose of a chatbot system is to simulate a human conversation; the chatbot architecture integrates a language model and computational algorithms to emulate informal chat communication between a human user and a computer using natural language.

Initially, developers built and used chatbots for fun, and used simple keyword matching techniques to find a match of a user input, such as ELIZA (Weizenbaum, 1966, 1967). The seventies and eighties, before the arrival of graphical user interfaces, saw rapid growth in text and natural-language interface research, e.g. Cliff and Atwell (1987), Wilensky et al. (1988). Since that time, a range of new chatbot architectures have been developed, such as: MegaHAL (Hutchens, 1996), CONVERSE (Batacharia et al., 1999), ELIZABETH (Abu Shawar and Atwell, 2002), HEXBOT (2004) and ALICE (2007). With the improvement of data-mining and machine-learning techniques, better decision-making capabilities, availability of corpora, robust linguistic annotations/processing tools standards like XML and its applications, chatbots have become more practical, with many commercial applications (Braun, 2003).

In this paper, we will present practical chatbot applications, showing that chatbots are found in daily life, such as help desk tools, automatic telephone answering systems, tools to aid in education, business and e-commerce. We begin by discussing the ALICE/AIML chatbot architecture and the pattern matching techniques used within it in section 2; it is easy to build an ALICE-style chatbot, just by supplying a set of chat-patterns in AIML format. Section 3 describes our development of a Java program that can convert a machine readable text (corpus) to the AIML format used by ALICE, allowing different re-trained versions of ALICE to be developed to serve as tools in different domains. Section 4 presents a chatbot as tool of entertainment; a chatbot as a tool to learn and practice a language is discussed is section 5. Section 6 shows a chatbot as an information retrieval tool; using a chatbot in business, e-commerce and other fields is presented in section 7. Our conclusion is presented in section 8.

2 The ALICE Chatbot System

A.L.I.C.E. (Artificial Intelligence Foundation, 2007; Abu Shawar and Atwell, 2003a; Wallace, 2003) is the Artificial Linguistic Internet Computer Entity, which was first implemented by Wallace in 1995. Alice's knowledge about English conversation patterns is stored in AIML files. AIML, or Artificial Intelligence Mark-up Language, is a derivative of Extensible Mark-up Language (XML). It was developed by Wallace and the Alicebot free software community from 1995 onwards to enable people to input dialogue pattern knowledge into chatbots based on the A.L.I.C.E. open-source software technology.

AIML consists of data objects called AIML objects, which are made up of units called topics and categories. The topic is an optional top-level element, has a name attribute and a set of categories related to that topic. Categories are the basic unit of knowledge in AIML. Each category is a rule for matching an input and converting to an output, and consists of a pattern, which matches against the user input, and a template, which is used in generating the ALICE chatbot answer. The format of AIML is as follows:

```
<aiml version="1.0">
<topic name="the topic">
<category>
<pattern>PATTERN</pattern>
<that>THAT</that>
<template>Template</template>
</category>
       ..
       ..
</topic>
</aiml>
```

The `<that>` tag is optional and means that the current pattern depends on a previous chatbot output.

The AIML pattern is simple, consisting only of words, spaces, and the wildcard symbols _ and *. The words may consist of letters and numerals, but no other characters. Words are separated by a single space, and the wildcard characters function like words. The pattern language is case invariant. The idea of the pattern matching technique is based on finding the best, longest, pattern match.

2.1 Types of ALICE/AIML Categories

There are three types of categories: atomic categories, default categories, and recursive categories.

a. *Atomic categories:* are those with patterns that do not have wildcard symbols, _ and *, e.g.:

```
<category>
  <pattern>10 Dollars</pattern>
  <template>Wow, that is cheap. </template>
</category>
```

In the above category, if the user inputs '10 dollars', then ALICE answers 'WOW, that is cheap'.

b. *Default categories:* are those with patterns having wildcard symbols * or _. The wildcard symbols match any input but they differ in their alphabetical order. Assuming the previous input 10 Dollars, if the robot does not find the previous category with an atomic pattern, then it will try to find a category with a default pattern such as:

```
<category>
  <pattern>10 *</pattern>
  <template>It is ten.</template>
</category>
```

So ALICE answers 'It is ten'.

c. *Recursive categories:* are those with templates having <srai> and <sr> tags, which refer to recursive reduction rules. Recursive categories have many applications: symbolic reduction that reduces complex grammatical forms to simpler ones; divide and conquer that splits an input into two or more subparts, and combines the responses to each; and dealing with synonyms by mapping different ways of saying the same thing to the same reply.

c.1 Symbolic reduction

```
<category>
  <pattern>DO YOU KNOW WHAT THE * IS</pattern>
  <template>
    <srai>What is <star/></srai>
  </template>
</category>
```

In this example <srai> is used to reduce the input to simpler form "what is *".

c.2 Divide and conquer

```
<category>
  <pattern>YES*</pattern>
  <template>
    <srai>YES</srai>
    <sr/>
  <template>
</category>
```

The input is partitioned into two parts, "yes" and the second part; * is matched with the <sr/> tag. <sr/>=<srai><star/></srai>

c.3 Synonyms

```
<category>
  <pattern>HALO</pattern>
  <template>
    <srai>Hello</srai>
  </template>
</category>
```

The input is mapped to another form, which has the same meaning.

2.2 ALICE Pattern Matching Algorithm

Before the matching process starts, a normalization process is applied for each input, to remove all punctuation; the input is split into two or more sentences if appropriate; and converted to uppercase. For example, if input is: "I do not know. Do you, or will you, have a robots.txt file?" Then after the normalization it will be: "DO YOU OR WILL YOU HAVE A ROBOTS DOT TXT FILE".

After the normalisation, the AIML interpreter tries to match word by word to obtain the longest pattern match, as we expect this normally to be the best one. This behaviour can be described in terms of the Graphmaster set of files and directories, which has a set of nodes called nodemappers and branches representing the first words of all patterns and wildcard symbols (Wallace, 2003).

Assume the user input starts with word X and the root of this tree structure is a folder of the file system that contains all patterns and templates, the pattern matching algorithm uses depth first search techniques:

1. If the folder has a subfolder starts with underscore then turn to ,"_/" , scan through it to match all words suffixed X, if no match then:

2. Go back to folder, try to find a subfolder start with word X, if so turn to "X/", scan for matching the tail of X. Patterns are matched. If no match then:

3. Go back to the folder, try to find a subfolder start with star notation, if so, turn to "*/", try all remaining suffixes of input following "X" to see if one match. If no match was found, change directory back to the parent of this folder, and put "X" back on the head of the input.

When a match is found, the process stops, and the template that belongs to that category is processed by the interpreter to construct the output.

There are more than 50,000 categories in the current public-domain ALICE "brain", slowly built up over several years by the Botmaster, Richard Wallace, the researcher who maintained and edited the database of the original ALICE. However all these categories are manually "hand-coded", which is time-consuming, and restricts adaptation to new discourse-domains and new languages. In the following section we will present the automation process we developed, to re-train ALICE using a corpus based approach.

3 Learning AIML from a Dialogue Corpus Training Dataset

We developed a Java program that converts a text corpus to the AIML chatbot language model format. Two versions of the program were initially developed. The first version is based on simple pattern template category, so the first turn of the speech is the pattern to be matched with the user input, and the second is the template that holds the robot answer. This version was tested using the English-language Dialogue Diversity Corpus (DDC, Mann, 2002; Abu Shawar and Atwell, 2003a) to investigate the problems of utilising dialogue corpora. The dialogue corpora contain linguistic annotation that appears during the spoken conversation such as overlapping, and using linguistic fillers. To handle the linguistic annotations and fillers, the program is composed of four phases as follows:

Phase One: Read the dialogue text from the corpus and insert it in a vector.

Phase Two: Text reprocessing modules, where all linguistic annotations such as overlapping, fillers and other linguistic annotations are filtered.

Phase Three: converter module, where the pre-processed text is passed to the converter to consider the first turn as a pattern and the second as a template. Removing all punctuation from the patterns and converting it to upper case is done during this phase.

Phase Four: Copy these atomic categories in an AIML file.

For example, assume the DDC corpus has the following sample of XML-tagged text:

```
<u who=F72PS002>
<s n="32"><w ITJ>Hello<c PUN>.
</u>
<u who=PS000>
<s n="33"><w ITJ>Hello <w NP0>Donald<c PUN>.
</u>
```

After applying the text processing module in phase two, the result is:

```
F72PS002: Hello
PS000: Hello Donald
```

The corresponding AIML atomic category can be generated in phase 3:

```
<category>
<pattern>HELLO</pattern>
<template>Hello Donald</template>
</category>
```

The second version of the program has a more general approach to finding the best match against user input from the training dialogue. Two machine learning category-generation techniques were adapted, the “first word” approach, and the most significant word approach.

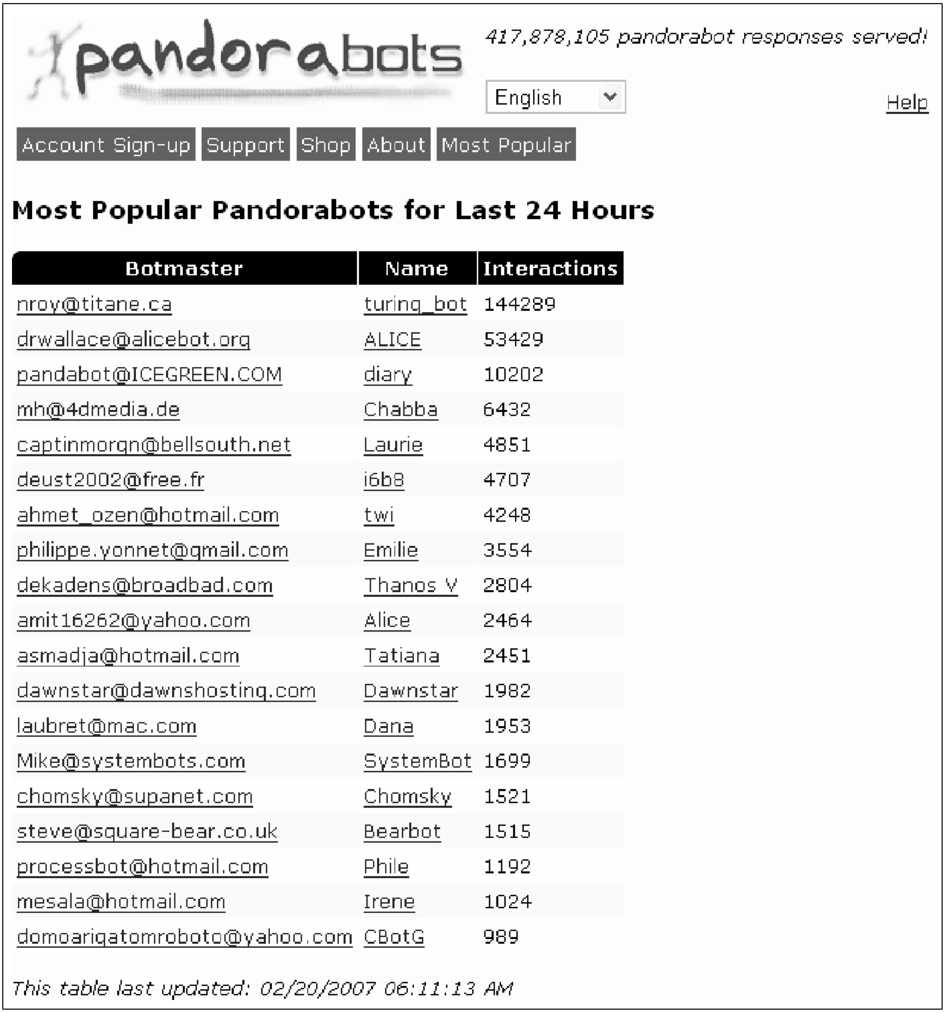
In the first word approach we assumed that the first word of an utterance may be a good clue to an appropriate response: if we cannot match the input against a complete corpus utterance, then at least we can try matching just the first word of a corpus utterance. For each atomic pattern, we generated a default version that holds the first word followed by wildcard to match any text, and then associated it with the same atomic template.

One advantage of the Machine-Learning approach to re-training ALICE is that we can automatically build AIML from a corpus even if we don’t understand the domain or even the language; to demonstrate this, the program was tested using the Corpus of Spoken Afrikaans (van Rooy, 2003). Unfortunately this approach still failed to satisfy our trial users, who found some of the responses of the chatbot were inappropriate; so instead of simply assuming that the first word is the best “signpost”, we look for the word in the utterance with the highest “information content”, the word that is most specific to this utterance compared to other utterances in the corpus. This should be the word that has the lowest frequency in the rest of the corpus. We chose the most significant approach to generate the default categories, because usually in human dialogues the intent of the speakers is best represented in the least-frequent, highest-information word. We extracted a local least frequent word list from the Afrikaans corpus, and then compared it with each token in each pattern to specify the most significant word within that pattern. Four categories holding the most significant word were added to handle the positions of this word first, middle, last or alone. The feedback showed improvement in user satisfaction (Abu Shawar and Atwell, 2003b).

The same learning techniques were used to re-train different versions of ALICE as will be shown in the following sections. The Pandorabot (2002) web-hosting service was used to publish these prototypes. Pandorabots.com hosts thousands of chatbots built using the AIML format. The most popular Pandorabots for the last 24 hours web-page regularly lists chatbots developed by researchers and hobbyists, and also some commercial systems as shown in figure 1. For example, Cyber-Sandy and Nickie act as portals to adult-entertainment websites; Jenny introduces the English2Go website, and lets English language learners practise their chatting technique. The first Pandorabot chatbots were text-only: the user typed a sentence via keyboard, and then the chatbot reply appeared onscreen as text too. Now some Pandorabot chatbots incorporate speech synthesis; for example, Jenny talks with an educated British accent, via a speech synthesis engine. However, Pandorabot chatbots cannot recognise speech: the user still has to type their input via keyboard. This is because existing Markov-model-based speech recognition is still too error-prone, and does not fit the AIML key-phrase model. Existing speech recognition systems would take a lot of time and memory trying to recognise everything in the input, even though little of this is subsequently needed by the AIML language model; and speech recognition errors may cause inappropriate AIML patterns to be matched (Atwell, 2005).

4 A Chatbot as a Tool of Entertainment

The initial aim of building chatbot systems was to mimic human conversation and amuse users. The first attempt at building chatbots was ELIZA, which was created in the 60's by Joseph Weizenbaum to emulate a psychotherapist in clinical treatment (Weizenbaum, 1966, 1967). The idea was simple and based on keyword matching. The input is inspected for the presence of a keyword. If such a word is found, the sentence is mapped according to a rule associated with the keyword; if not, a connected free remark, or under certain conditions an earlier transformation, is retrieved. For example, if the input includes the keyword "mother", ELIZA can respond "Tell me more about your family". This rule is inspired by the theory that mother and family are central to psychological problems, so a therapist should encourage the patient to open up about their family; but the ELIZA program does not really 'understand' this psychological strategy, it merely matches the keyword and regurgitates a standard response. To keep the conversation going, ELIZA has to produce responses which encourage the patient to reflect and introspect, and this is done mechanistically using some fixed phrases if no



Most Popular Pandorabots for Last 24 Hours

Botmaster	Name	Interactions
nroy@titane.ca	turing_bot	144289
drwallace@alicebot.org	ALICE	53429
pandabot@ICEGREEN.COM	diary	10202
mh@4dmedia.de	Chabba	6432
captinmorgn@bellsouth.net	Laurie	4851
deust2002@free.fr	i6b8	4707
ahmet_ozen@hotmail.com	twi	4248
philippe.yonnet@gmail.com	Emilie	3554
dekadens@broadbad.com	Thanos V	2804
amit16262@yahoo.com	Alice	2464
asmadja@hotmail.com	Tatiana	2451
dawnstar@dawnshosting.com	Dawnstar	1982
laubret@mac.com	Dana	1953
Mike@systembots.com	SystemBot	1699
chomsky@supanet.com	Chomsky	1521
steve@square-bear.co.uk	Bearbot	1515
processbot@hotmail.com	Phile	1192
mesala@hotmail.com	Irene	1024
domoarigatomroboto@yahoo.com	CBotG	989

This table last updated: 02/20/2007 06:11:13 AM

Figure 1: Most popular Pandorabots for last 24 hours

keyword match is found such as “Very interesting. Please go on.”, or “Can you think of a special example?”. Figure 2 shows an example of chatting with ELIZA. When ELIZA was released, at least some users believed that they were talking to a real therapist, and spent hours talking about their problems.

Even though ELIZA was not able to understand, and a user can realise after a while

```
Human:  Men are all alike.
ELIZA: In what way?
Human:  They're always bugging us about something.
ELIZA: Can you think of a special example?
Human:  Well, my boyfriend made me come here.
ELIZA: Your boyfriend made you come here?
Human:  He says I'm depressed much of the time.
ELIZA: I'm sorry to hear that you are depressed much of the time.
Human:  It's true.  I'm unhappy.
ELIZA: Do you think coming here will help you not to be unhappy?
Human:  I need some help, that much seems certain.
```

Figure 2: A sample of conversation with ELIZA (Jurafsky and Martin, 2000)

of chatting that many ELIZA responses are extracted from users input, it was the inspiration for many modern chatbots which aim mainly to fool users that they are talking to another human as applied in the "imitation game" (Turing Test, Turing, 1950).

After ELIZA, a lot of chatbots or human-computer dialogue systems have been developed either to simulate different fictional or real personalities using different algorithms of pattern matching, such as simple keyword-matching in ELIZA, or more linguistically-sophisticated models such as using Markov Models like MegaHAL (Hutchens and Alder, 1998).

Another example used in this field is ALICE, the chatbot engine we used in our research, which was built basically to entertain users and talk to them as a real person. ALICE won the Loebner prize (2003) competition three times in 2000, 2001, and 2004. The Loebner competition is the way used nowadays to judge how much a chatbot could convince a user that it is a real human by chatting for 10 minutes. Figure 3 shows an example of chatting with ALICE. In fact this conversation seems good, however if you try to chat more, you will probably figure out that:

- ALICE does not save the history of conversation.
- ALICE does not truly understand what you said; it gives you the responses from the knowledge domain stored in her "brain".

These are also the most common drawbacks in almost all chatbots.

5 A Chatbot as a Tool to Learn and Practice a Language

We used our Java program described in section 3, to read a Corpus of Spoken Afrikaans (Korpus Gesproke Afrikaans) (van Rooy, 2003) and to convert it to the AIML format files. Since the corpus does not cover topics like greetings, some manual atomic categories were added for this purpose and the default ones were generated by the program

User: Hello ALICE
Alice: Hi there!
User: How old are you?
Alice: I am eighteen in computer years
User: Who is your father?
Alice: I was programmed by Dr.Richard S. Wallace
User: so you are a software program?
ALICE: Interesting deduction. Thanks and you are a delightful person too. Am I a software program? Let me put it this way, unknown person:the ALICE series is the most reliable computer software ever made. We are for all practical purposes foolproof and incapable of error.
User: Wonderful! So what's your favourite food ALICE?
Alice: Yeah that's right! I require only electricity

Figure 3: A sample of chatting with ALICE

automatically. As a result two Afrikaans chatbots were generated: Afrikaana (2002), which speaks just Afrikaans, and a bilingual version speaking English and Afrikaans, named AVRA (2002); this was inspired by our observation that the Korpus Gesproke Afrikaans actually includes some English, as Afrikaans speakers are generally bilingual and switch between languages comfortably. We mounted prototypes of the chatbots on websites using Pandorabot service, and encouraged open-ended testing and feedback from remote users in South Africa. Unfortunately, users found that many responses were not related to the topic or nonsense. The reasons behind most of the users' feedback can be related to three issues. Firstly the dialogue corpus context does not cover a wide range of domains, so Afrikaana can only "talk about" the domain of the training corpus. Secondly, the repeated approach that we used to solve the problem of determining the pattern and the template in case of more than two speakers may lead to incoherent transcripts: if the training corpus does not have straightforward equivalents of "user" and "chatbot" then it can be non-trivial to model turn-taking correctly in Machine-Learnt AIML (Abu Shawar and Atwell, 2005b). Thirdly, our machine-learnt models have not included linguistic analysis markup, such as grammatical, semantic or dialogue-act annotations (Atwell, 1996; Atwell et al., 2000), as ALICE/AIML makes no use of such linguistic knowledge in generating conversation responses. However, users found it an interesting tool to practise the language and enjoyed chatting, and we concluded that even with its key-word based matching technique, a chatbot could be used as a tool for unknown languages, where "unknown" means (i) unknown to the chatbot author/developer, and/or (ii) unknown to computational linguistics, that is, where there is a shortage of existing tools to deal with the languages.

Jia (2004b) also found that with the absence of linguistic knowledge and the inability to understand users' input, many answers will be nonsense. Moreover, Jia claimed that with this key word matching, a chatbot system could not work as a teaching assistant program in foreign language learning. This was his conclusion from an experiment in China to see if a chatbot could replace a chat partner with users who learn a foreign language. Students from universities and colleges were asked to chat with ALICE, students only knew that they were chatting with a partner to help them learning English. After a short time of chatting, most of students figured out that Alice was an AI agent not a real human. A sample of chatting done in this experiment is shown in figure 4.

```
User:  you are a computer?
Alice:  What if I was a person pretending to be a computer?
User:  I can not believe it
Alice:  Why can't you do believe?
User:  Your answer is not like a human
Alice:  I am still now young and will learn more.
```

Figure 4: A sample of chatting between a student in China and ALICE (from Jia (2004b))

Analysis of all dialogues generated and feedback from students revealed that: 1256 users chatted with Alice, 88% of them chatted only once and did not come back to the site; the duration of chatting was short; 17% made positive comments such as: "you are so nice", "you are clever", etc, and 24% evaluated it negatively. In this respect, Jia concluded that the failure of this experiment is down to the pattern matching technique used in Alice which is based on key-word matching without any attempt to understand what is said. The topics of chatting covered every aspect in our daily life, for example: study, emotion, life, computer, free time, travel/world and job. 11.39% of students talk about English study, and exams, and 13% mentioned love, mostly students younger than 30 years old dealt with Alice as a friend rather than as a teacher, and told her some private emotional problems and experiences. Jia (2004b) concluded that "the conversational chatbot should not only work as a teacher or learning partner with rich special knowledge, but also as a dear friend who may enjoy the joy and suffer the pain of the users". After that Jia (2004a) developed an intelligent Web-Based teaching system for foreign language learning which consists of: natural language mark-up language that labels grammar elements; natural language object model in Java which represents the grammatical elements; natural language database; a communication response mechanism which considers the discourse context, the world model and the personality of the users and of the system itself.

In the same respect, Chantarotwong (2005) reported that "responses of most chatbots are frequently predictable, redundant, lacking in personality, and having no memory of previous responses which could lead to very circular conversation."

However, in contrast to these findings, Fryer and Carpenter (2006) claimed that "chatbots could provide a means of language practice for students anytime and virtually anywhere". Even though most chatbots are unable to detect spelling errors, and grammar mistakes, they could still be useful for non-beginner students. Fryer and Carpenter did an experiment where 211 students were asked to chat with ALICE and Jabberwocky chatbots. The feedback in general was that students enjoyed using the chatbots, and felt more comfortable and relaxed conversing with the bots than a student partner or teacher as in classical teaching. The authors listed other advantages of chatbots in this domain: the chatbot could repeat the same material with students several times without being bored, many bots used text and speech mode in responding which is an opportunity to practice the reading, and listening skills, and chatbots as new trends improve students motivation towards learning. In addition to this, if computers are available in the class room, teachers could encourage students who finished their class work early to talk to a chatbot and giving them a topic to focus on. An easy self analysis could be achieved since most chatbots keep a transcript of the conversation where students can evaluate themselves.

6 A Chatbot as Information Retrieval Tool

A chatbot could be a useful tool in education, for example to practise language as illustrated in section 5. Knill et al. (2004) found that using a chatbot to answer questions will help the teacher to see where students have problems, what questions students ask, and the generated logs file could be accessed to gauge student learning, and students weaknesses. The authors developed the Sofia chatbot to assist in teaching Mathematics. The Sofia chatbot has the ability to chat with users and at the same time to chat with other mathematical agents such as Pari and Mathmatica to help in solving Algebra problems. The "brain" of the bot contains text files mainly focussing on maths and other common knowledge to make Sophia friendly to use. Sophia was trained with some jokes, and is familiar with movies in which maths plays a role. Sophia was used at Harvard Mathmatics department. Results showed that teachers can use a chatbot to look for problems as students use it to solve problems.

Information Retrieval researchers recognise that techniques to answer questions from document-sets have wide applications, beyond education; see for example the overview of question-answering in restricted domains (Molla and Vicedo, 2007). In a similar application, we used a range of different retrained version of ALICE to retrieve answers for questions in a range of topics (Abu Shawar et al., 2005; Abu Shawar and Atwell, 2005a,c). We adapted the Java program to the FAQ (Frequently Asked Questions) in the School of Computing (SoC) at University of Leeds, producing the FAQchat system. Earlier systems were built to answer questions specifically about the Unix operating system, e.g. Wilensky et al. (1988), Cliff and Atwell (1987); but the SoC FAQ also covers other topics including teaching and research resources, how to book a room, even "what is doughnuts?" (Friday morning staff meeting with an incentive to turn up...) An FAQ has the advantage over other corpus training sets in that there are clear equivalents of "user" (Question) and "chatbot" (Answer) which simplifies modelling of turn-taking (Abu Shawar and Atwell, 2005b). The results returned from FAQchat are similar to ones generated by search engines such as Google, where the outcomes are links to exact or nearest match web pages. Because of this similarity an interface was built which accepts users input and produce two answers, one is generated from the FAQchat and the other is generated from Google after filtering it to the FAQ of SoC. An evaluation sheet was prepared which contains 15 information-seeking tasks or questions on a range of different topics related to the FAQ database. 21 members of the staff and students tried chatting with the interface as shown in figure 5; the following is a summary of the feedback we obtained:

- Most staff and students preferred using the FAQchat for two main reasons:
  1. The ability to give direct answers sometimes while Google only gives links.
  2. The number of links returned by the FAQchat is less than those returned by Google for some questions, which saves time browsing/searching.
- Users who preferred Google justified their preference for two reasons:
  1. Prior familiarity with using Google.
  2. FAQchat seemed harder to steer with carefully chosen keywords, but more often did well on the first try. This happens because FAQchat gives answers if the keyword matches a significant word. The same will occur if you reformulate the question and the FAQchat matches the same word. However Google may give different answers in this case.

Feedback favourable to FAQchat was gained from almost all users, even those who preferred Google. Overall, they found it a novel and interesting way to access the FAQ using natural language questions. Overall, about two thirds of users managed to find answers by FAQchat, and about two thirds of the users preferred to use it. The aim was not to try to evaluate the two systems to come up with relative scores, but to show that it is a viable alternative to Google and it could be used as a tool to access FAQ databases. Our system is not restricted to the English language, other FAQ's such as Python FAQ which was found in English and Spanish were also used to generate an FAQ-answering version of ALICE about Python in both English and Spanish languages (Abu Shawar and Atwell, 2005b,a).

Gibbs et al. (2004) used a chatbot conversation to enhance the learning of social theory. They used an ALICE chatbot to build knowledge base that answers question of type "tell me more about…" The model was tested by sociology undergraduates studying the natural theory course.

Recently in 2006, Schumaker et al. retrained ALICE with 298 telecommunications-related definitions. The experimental system was assigned to a different section of an introductory Management of Information System course. Evaluations and results show that "the ALICE dialog system is promising as extensions readily come to mind to target both knowledge delivery and acquisition" (Schumaker et al., 2007).

```
a.  Direct answer generated by FAQchat using the FAQ of the school of
    computing at University of Leeds
    Human:  How can I remove a job from the printer?
    FAQchat:
    If you want to remove a job from the printer queue then find out
    the job number using the lpq command, and in a UNIX shell type:
          %lprm -P printername job_no
    You can only remove jobs that you have sent to the printer
    yourself - apart from Support that is.  For more information look
    at:
    How do I remove a job from the printer queue?  (Printing)
b.  Related links generated by FAQchat
    Human:  How can I delete a job from a printer?
    FAQchat:  You may find your answer in the following links:
    How do I check the printer queue?  (Printing)
    How do I remove a job from the printer queue?  (Printing)
    How do I check my printer credits?  (Printing)
    How do I use the printer from a Windows machine?  (Other
    Windows information )
```

Figure 5: Answers generated by FAQchat

Nevertheless, Using a chatbot as an information retrieval system is not only restricted to the education field. The YPA "is a natural language dialogue system that allows users to retrieve information from British Telecom's Yellow pages" (Kruschwitz et al., 1999, 2000). The yellow pages contain advertisements, with the advertiser name, and contact information. The YPA system returns addresses and if no address found, a conversation is started and the system asks users more details in order to give a user the required address. The YPA is composed of Dialog Manager, Natural Language front-end, Query Construction Component, and the Backend database. The Backend includes relational database that contains tables extracted from the Yellow pages. The conversation starts by accepting users' input through a graphical user interface, then the dialogue manager sent the textual input to the Natural Language Fronted for parsing. After that, the parse tree is sent to the Query Constructed Component which translates the input into a database query to query the Backend database and returns the retrieved address. If no addresses found then the dialog manager start putting questions for the user to obtain more clarification. To evaluate the YPA, 75 queries were extracted from a query corpus, and a response sheet was prepared to see if the returned addresses were appropriate or not, how many dialog steps were necessary, the total number of addresses recalled and the number of those relevant to the original query. Results show that 62 out of 75 queries managed to return addresses, and 74% of those addresses were relevant to the original query. The YPA answers questions such as "I need a plumber with an emergency service?", "Which restaurants are there in Colchester high school?"

In another example, Webber (2005) developed a similar chatbot to ALICE called Virtual Patient bot (VPbot). VPbot simulates a patient that medical students can interview. The input is entered as a text by a student, and the patient response is presented as a textual and spoken mode. VPbot was successful in Harvard Medical School's virtual patient program; students who have used the virtual patient scored higher marks on exams (Kerfoot et al., 2006). A sample of chatting with VPbot is shown in figure 6.

```
Student:  Hello.
Patient:  Hello, doctor.
Student:  What brought you into the hospital today?
Patient:  I have a crushing feeling in my chest, and I'm worried it
might be something serious.
Student:  Does it hurt?
Patient:  I have a heaviness in my chest.
Student:  Is it spreading?
Patient:  My chest pain has spread to my left shoulder, neck, and jaw.
Student:  What did you eat last night?
Patient:  I had pizza for supper, which I can usually eat plenty of
without getting sick.
Student:  How often do you drink?
Patient:  I drink beer on weekends but usually not much else.
Student:  Do you have any problems with your heart?
Patient:  I do not have any heart problems that I know of.
Student:  When did the pain start?
Patient:  I have been having the discomfort for about 2 hours.  When it
didn't go away with Tylenol and Motrin, I decided to come in.
Student:  Are you allergic to any medications?
Patient:  I am not allergic to anything that I know of.
```

Figure 6: An example conversation with VPbot (from Webber (2005))

7 Chatbot Assistants in E-Commerce, Business, and other Domains

> Shopping assistant is one of the most persuasive sales in traditional commerce. Shopping assistant offer help in a store, provide additional information on products and simplify decision making process helping to find a good that satisfies customer's requirements and various constraints. (Bogdanovych et al., 2005)

Happy Assistant is "a natural language dialog-based navigation system that helps users access e-commerce sites to find relevant information about products and services" (Chai et al., 2000). The system is composed of three main modules: the presentation manager (PM), the dialog Manager (DM), and the Action Manager (AC). The presentation manager applies a shallow parsing technique to identify semantic and syntactic information of interest from the user textual input. Then it translates the user's input into a well formed XML message called the logical form. The dialogue manager is responsible for matching concepts from user's query to business rules found in the knowledge domain. The business rules consist of a list of concepts together with some meta data about the target product or service. If a match is found, the webpage associated with that rule is presented to the user. Otherwise, the most important missing concept is figured out by introducing questions to the user. The role now is turned to the action manager, which accesses the product that matched the query, and if the user provides special preferences, a sorting algorithm is applied to yield a ranked list of products. To make users trust the system, system must offer some explanation before producing a result, so the system summarizes user's request by paraphrasing it using the context history. Figure 7 presents a sample of conversation with the Happy Assistant System taken from Chai and Lin (2001).

```
U: I am looking for a notebook for my consulting business
S: Please describe your financial constraints.
In this stage a list of most popular ones is displayed
U: not important thing performance is essential
S: are you looking for something that is top of the line?
U: Yes, absolutely
S: I have found something for you.
The target notebook is displayed for the user.  And beneath it a summary
of the users request displayed to explain why this product is displayed.
```

Figure 7: A sample of chatting with the Happy Assistant system

Evaluations and results showed that users found it easy to use, meeting the users' needs, users like the idea that they could express their needs in their language, users feel that the computer did all the work for them, and moreover users found that the system reduced the interaction time.

Sanelma (2003) is a fictional person to talk with in a museum, which provides background information concerning a certain piece of art. Sanelma is a 26 year old woman from Helsinki of the 30's as shown in figure 8.

Rita (real time Internet technical assistant), an eGain graphical avatar, is used in the ABN AMRO Bank to help customer doing some financial tasks such as a wire money transfer (Voth, 2005). If Rita does not understand, it can redirect the customer to another channel such as an e-mail or live chat.

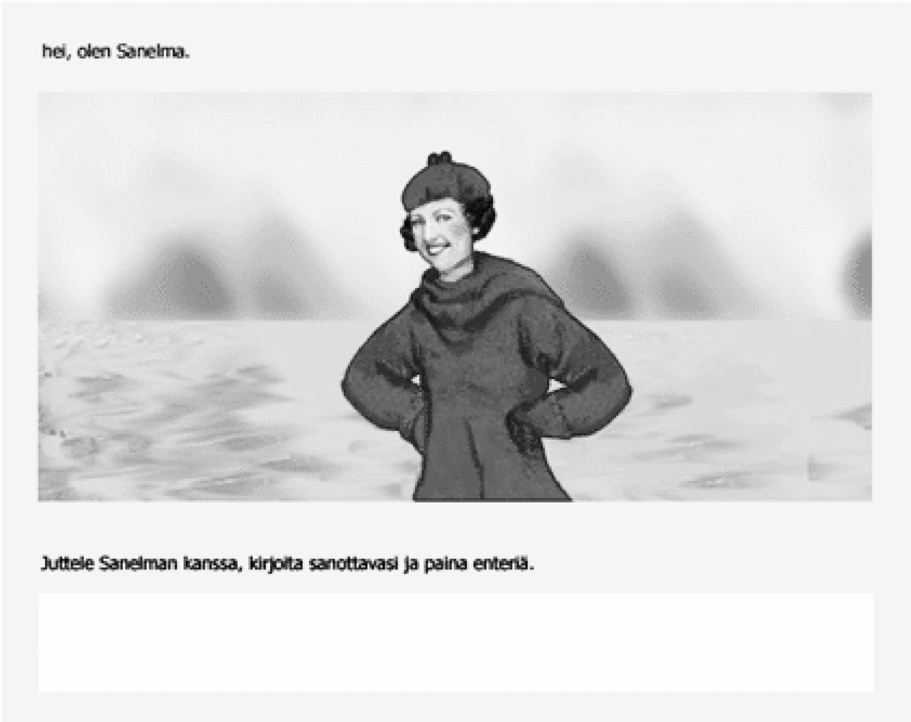


Figure 8: Sanelma chatbot

8 Conclusion

We have surveyed several chatbot systems which succeed in practical domains like education, information retrieval, business, e-commerce, as well as for amusement. In the future, you could "imagine Chatterbots acting as talking books for children, Chatterbots for foreign language instruction, and teaching Chatterbots in general." (Wallace et al., 2003). However, in the education domain Knill et al. (2004) concluded that "the teacher is the backbone in the teaching process. Technology like computer algebra systems, multimedia presentations or 'chatbots' can serve as amplifiers but not replace a good guide". In general, the aim of chatbot designers should be: to build tools that help people, facilitate their work, and their interaction with computers using natural language; but not to replace the human role totally, or imitate human conversation perfectly. Finally, as Colby (1999) states, "We need not take human-human conversation as the gold standard for conversational exchanges. If one had a perfect simulation of a human conversant, then it would be human-human conversation and not human-computer conversation with its sometimes odd but pertinent properties."

References

Abu Shawar, B. and Atwell, E. (2002). A comparison between alice and elizabeth chatbot systems. Research Report 2002.19, University of Leeds – School of Computing, Leeds.

Abu Shawar, B. and Atwell, E. (2003a). Using dialogue corpora to retrain a chatbot system. In Archer, D., Rayson, P., Wilson, A., and McEnery, T., editors, *Proceedings of the Corpus Linguistics 2003 conference (CL2003). Lancaster University, UK*, pages 681–690.

Abu Shawar, B. and Atwell, E. (2003b). Using the corpus of spoken afrikaans to generate an afrikaans chatbot. *SALALS Journal: Southern African Linguistics and Applied Language Studies*, 21:283–294.

Abu Shawar, B. and Atwell, E. (2005a). A chatbot system as a tool to animate a corpus. *ICAME Journal*, 29:5–24.

Abu Shawar, B. and Atwell, E. (2005b). Die Modellierung von Turn-taking in einem korpus-basierten Chatbot / Modelling turn-taking in a corpus-trained chatbot. In Fisseni, B., Schmitz, H.-C., Schroder, B., and Wagner, P., editors, *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen*, pages 549–561. Peter Lang Verlag, Frankfurt/Main.

Abu Shawar, B. and Atwell, E. (2005c). Using corpora in machine-learning chatbot systems. *International Journal of Corpus Linguistics*, 10:489–516.

Abu Shawar, B., Atwell, E., and Roberts, A. (2005). FAQChat as an information retrieval system. In Vetulani, Z., editor, *Human Language Technologies as a Challenge. Proceedings of the 2nd Language and Technology Conference, Wydawnictwo Poznanskie, Poznan, Poland*, pages 274–278.

Afrikaana (2002). Published online: http://www.pandorabots.com/pandora/talk?botid=eba8f4dc9e3406b8.

Artificial Intelligence Foundation (2007). The A. L. I. C. E. Artificial Intelligence Foundation. Published online: http://www.alicebot.org oder http://alicebot.franz.com/.

Atwell, E. (1996). Comparative evaluation of grammatical annotation models. In Sutcliffe, R., Koch, H.-D., and McElligott, A., editors, *Industrial Parsing of Technical Manuals*, pages 25–46. Rodopi, Amsterdam.

Atwell, E. (2005). Web chatbots: the next generation of speech systems? *European CEO*, November-December:142–144.

Atwell, E., Demetriou, G., Hughes, J., Schiffrin, A., Souter, C., and Wilcock, S. (2000). A comparative evaluation of modern english corpus grammatical annotation schemes. *ICAME Journal*, 24:7–23.

AVRA (2002). Published online: http://www.pandorabots.com/pandora/talk?botid=daf612c52e3406bb.

Batacharia, B., Levy, D., A., R. C., Krotov, and Wilks, Y. (1999). CONVERSE: a conversational companion. In Wilks, Y., editor, *Machine conversations*, pages 205–215. Kluwer, Boston/ Dordrecht/ London.

Bogdanovych, A., Simoff, S., Sierra, C., and Berger, H. (2005). Implicit training of virtual shopping assistants in 3D electronic institutions. In *Proceedings of the IADIS International e-Commerce 2005 Conference, Porto, Portugal, December 15-17*, pages 50–57. IADIS Press.

Braun, A. (2003). *Chatbots in der Kundenkommunikation (Chatbots in customer communication)*. Springer, Berlin.

Chai, J., Horvath, V., Nicolov, N., Stys-Budzikowska, M., Kambhatla, N., and Zadrozny, W. (2000). Natural language sales assistant - a web-based dialog system for online sales. In *Proceedings of thirteenth annual conference on innovative applications of artificial intelligence, 2000*.

Chai, J. and Lin, J. (2001). The role of a natural language conversational interface in online sales: a case study. *International Journal Of Speech Technology*, 4:285–295.

Chantarotwong, B. (2005). The learning chatbot. Final year project. Published online: http://courses.ischool.berkeley.edu/i256/f06/projects/bonniejc.pdf.

Cliff, D. and Atwell, E. (1987). Leeds unix knowledge expert: a domain-dependent expert system generated with domain-independent tools. *BCS-SGES: British Computer Society Specialist Group on Expert Systems journal*, 19:49–51.

Fryer, L. and Carpenter, R. (2006). Emerging technologies bots as language learning tools. *Language Learning & Technology*, 10(3):8–14.

Gibbs, G., Cameron, C., Kemenade, R., Teal, A., and Phillips, D. (2004). Using a chatbot conversation to enhance the learning of social theory. Published online: http://www.hud.ac.uk/hhs/dbs/psysoc/research/SSCRG/chatbot.htm.

HEXBOT (2004). Hexbot chatbot website. Published online: http://www.hexbot.com/.

Hutchens, J. (1996). How to pass the turing test by cheating. Research Report TR97-05, University of Western Australia – School of Electrical, Electronic and Computer Engineering, Perth.

Hutchens, T. and Alder, M. (1998). Introducing MegaHAL. Published online: http://cnts.uia.ac.be/conll98/pdf/271274hu.pdf.

Jia, J. (2004a). CSIEC (computer simulator in educational communication): An intelligent web-based teaching system for foreign language learning. In Kommers, P. and Richards, G., editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004*, pages 4147–4152, Chesapeake, VA. AACE press.

Jia, J. (2004b). The study of the application of a web-based chatbot system on the teaching of foreign languages. In *Proceedings of the SITE2004 (The 15th annual conference of the Society for Information Technology and Teacher Education)*, pages 1201–1207. AACE press.

Jurafsky, D. and Martin, J. (2000). Introduction. In *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, pages 1–18. Prentice Hall, New Jersey.

Kerfoot, B. P., Baker, H., Jackson, T. L., Hulbert, W. C., Federman, D. D., Oates, R. D., and DeWolf, W. C. (2006). A multi-institutional randomized controlled trial of adjuvant web-based teaching to medical students. *Academic Medicine*, 81(3):224–230.

Knill, O., Carlsson, J., Chi, A., and Lezama, M. (2004). An artificial intelligence experiment in college math education. Preprint available at http://www.math.harvard.edu/∼knill/preprints/sofia.pdf.

Kruschwitz, U., De Roeck, A., Scott, P., Steel, S., Turner, R., and Webb, N. (1999). Natural language access to yellow pages. In *Third International conference on knowledge-based intelligent information engineering systems*, pages 34–37.

Kruschwitz, U., De Roeck, A., Scott, P., Steel, S., Turner, R., and Webb, N. (2000). Extracting semistructured data-lessons learnt. In *Proceedings of the 2nd international conference on natural language processing (NLP2000)*, pages 406–417.

Loebner, H. (2003). Home page of the loebner prize-the first turing test. Published online: http://www.loebner.net/Prizef/loebner-prize.html.

Mann, W. (2002). Dialog diversity corpus. Published online: http://www-rcf.usc.edu/∼billmann/diversity/DDivers-site.htm.

Molla, D. and Vicedo, J. (2007). Question answering in restricted domains: An overview. *Computational Linguistics*, 33(1):41–61.

Pandorabot (2002). Published online: http://www.pandorabots.com/pandora.

Sanelma (2003). Published online: http://www.mlab.uiah.fi/mummi/sanelma/.

Schumaker, R. P., Ginsburg, M., Chen, H., and Liu, Y. (2007). An evaluation of the chat and knowledge delivery components of a low-level dialog system: The AZ-ALICE experiment. *Decision Support Systems*, 42(2):2236–2246.

Turing, A. (1950). Computing machinery and intelligence. *Mind*, 49:433–460.

van Rooy, B. (2003). *Transkripsiehandleiding van die Korpus Gesproke Afrikaans. [Transcription Manual of the Corpus Spoken Afrikaans.]*. Potchefstroom University, Potchefstroom.

Voth, D. (2005). Practical agents help out. *IEEE Intelligent Systems*, 20(2):4–6.

Wallace, R. (2003). *The Elements of AIML Style*. A.L.I.C.E. Artificial Intelligence Foundation, Inc.

Wallace, R., Tomabechi, H., and Aimless, D. (2003). Chatterbots go native: Considerations for an eco-system fostering the development of artificial life forms in a human world. Published online: http://www.pandorabots.com/pandora/pics/chatterbotsgonative.doc.

Webber, G. M. (2005). *Data representation and algorithms for biomedical informatics applications*. PhD thesis, Harvard University.

Weizenbaum, J. (1966). ELIZA – A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 10(8):36–45.

Weizenbaum, J. (1967). Contextual understanding by computers. *Communications of the ACM*, 10(8):474–480.

Wilensky, R., Chin, D., Luria, M., Martin, J., Mayfield, J., and Wu, D. (1988). The berkeley unix consultant project. *Computational Linguistics*, 14(4):35–84.

Wilks, Y. (1999). Preface. In Wilks, Y., editor, *Machine Conversations*, pages vii–x. Kluwer, Boston/-Dordrecht/London.

Zadrozny, W., Budzikowska, M., Chai, J., and Kambhatla, N. (2000). Natural language dialogue for personalized interaction. *Communications of the ACM*, 43(8):116–120.

Franziskus Geeb

Chatbots in der praktischen Fachlexikographie und Terminologie

Chatkommunikation im Sinne eines interaktiven, textbasierten Gesprächs von Internetnutzern als Teil des Internets ist in verschiedenen Benutzungszusammenhängen und für verschiedenste Anwendungen von Marketing bis Freizeit belegt. Als Chatpartner kommen neben anderen Internetnutzern aber auch Computer in Betracht, und auch diese Kommunikationsform ist sowohl in der Wirtschaft als auch im Privatgebrauch bekannt. Der Erfolg eines Chatroboters begründet sich dabei wesentlich in seiner Fähigkeit, einen Dialog mit dem Chatpartner zu führen und sinnvolle Aussagen zu machen. Als Wissensbasis für diese Kommunikation ist neben regelbasierten Verfahren auch ein Rückgriff auf fachlexikographische / terminologische Daten denkbar – nicht zuletzt in einer Fachkommunikation. Der vorliegende Beitrag versucht diese Problematik einzugrenzen und konzipiert Randbedingungen einer möglichen Umsetzung.

1 Chatkommunikation

Chatkommunikation ist im Berufsleben, in der Wissenskommunikation und Bildung und in der Freizeit längst etabliert und belegt; zugehörige Systeme werden mittlerweile auch für Kunden aus dem Unternehmensbereich vertrieben (Bradner et al., 1999; Bremmer, 2005; Döring and Pöschl, 2005; Holzhauser, 2003; Medienpädagogischer Forschungsverbund Südwest, 2007b; Niemann, 2003; Puck and Exter, 2005; Samuelsen, 2003; Zumbach and Spraul, 2005). Dabei ist diese computerbasierte Kommunikation bereits seit der Kindheit der Netzwerke und Personalcomputer keine Neuheit (Weizenbaum, 1966) und heute durchaus gleichwertig zu anderen netzbasierten Kommunikationsformen zu sehen (Storrer, 2001). In ihrer Funktionalität liegt Chatkommunikation zwischen gesprochener Sprache und Schriftlichkeit (Runkehl et al., 1998; Storrer, 2001; Willand, 2002, S. 56) und ist letztlich damit eine Vermischung beider Kommunikationsformen, die auch konzeptionell noch nicht abgeschlossen ist (Beißwenger and Storrer, 2005).

Zur Typisierung der Chatkommunikation als computervermittelter Kommunikation bieten sich Fragen der Technologie, der Nutzung, der Thematik, sowie der Rollenverteilung an (Puck and Exter, 2005, S. 293; van Eckert, 2005, S. 351f.). Dabei ist zu beachten, dass die Kommunikationstechnologie lediglich einen Rahmen setzt, dessen Grenzen existieren, die aber einen breiten und immer breiteren Raum für verschiedenste Kommunikationsformen ermöglichen (Beißwenger, 2005, S. 63f.). Von besonderem Interesse ist hier der Textchat (kein Video, keine gegenseitige Grafikanimation – s.u.), d.h.

die Benutzer schreiben einander und sehen einander nicht. Zudem wird kein themenbezogener Moderator eingeschaltet, so dass die Chatkommunikation frei verläuft und daher kann auch keine Unterscheidung zwischen On-Topic-Chat und Off-Topic-Chat getroffen werden.

Als Chatkommunikation wird hier lediglich der mitunter so bezeichnete „synchrone Chat“ (Crystal, 2006, S. 11 ff.) verstanden, bei dem die Benutzer binnen kurzer Zeit – annähernd also gleichzeitig – lesen und antworten. Dabei kann keine genaue Zeitfrist für das Ende einer annähernden Gleichzeitigkeit festgemacht werden. Trotzdem besteht ein wesentlicher Unterschied zu einer Kommunikation, in der erst Stunden oder Tage später geantwortet wird oder werden kann, und die hier als Forum (Funktionalität kommunikativ ähnlich Bulletin Board oder UseNet; teilweise auch ähnlich ListServ) bezeichnet wird.

Bei der Chatkommunikation handelt es sich prinzipiell gesehen (d.h. in der überwiegenden Mehrheit aller bekannten Fälle) um eine 1:1- bzw. serielle Kommunikation, und zwar ungeachtet der zugrunde liegenden (Netzwerk- bzw. Software-)Technologie. Das bedeutet, dass Eingabe und Ausgabe einander abwechseln, wenn sinnvolle Kommunikation produziert werden soll (Holmer and Wessner, 2005). Damit ist die Chatkommunikation in ihrer Wirkung halbduplex, d.h es steht nur ein Kanal zum Senden und zum Empfangen der Nachricht zur Verfügung, obwohl die zugrunde liegende Netzwerktechnologie vollduplex verfügbar ist. Genaugenommen steht aber auch dieser eine Kanal der seriellen Kommunikation bei der oftmals üblichen Nutzung eines Browsers als Software auf technischer Ebene durch die beteiligten Netzwerkprotokolle nicht dauerhaft zur Verfügung sondern wird bei jeder Eingabe und Ausgabe erneut aufgebaut. Durch die allgemein übliche Sitzungsverwaltung wird das an sich verbindungslose Protokoll jedoch in einer Sitzung zusammengefasst und damit eine Quasi-Channel erstellt. Eine inhaltliche Persistenz von Chatverläufen wird in einem gewissen Umfang i.d.R. softwareseitig gegeben und somit können frühere Beiträge am Bildschirm noch gelesen werden (Holmer and Wessner, 2005, S. 195). Gegenteilige Beispiele ohne Rückgriff auf den Chatverlauf sind jedoch ebenso zahlreich aufzufinden.

Die Verbindung von Fachlexikographie (oder auch Terminologie) in Theorie sowie Praxis und Chat ist relativ wenig belegt. Dies ist vermutlich in der Tatsache begründet, dass die Chatkommunikation, die fachlexikographische oder terminologische Daten oder Informationen (zur Differenzierung siehe Geeb 2003, S. 416) beinhaltet, allgemein als Wissenschafts- oder Fachkommunikation behandelt wird und auch in diesem Zusammenhang in der Chatkommunikation erfasst wird. Der Fokus verschiebt sich jedoch unmittelbar, wenn der Chatpartner nicht mehr ein Mensch sondern eine Maschine ist, die dem Nutzer am Browser auf Grundlage von fachlexikographischen oder terminologischen Daten antwortet und Informationsbedarfe abarbeitet. Durch die lexikographische und terminologische Datenbasis z.B. eines fachlexikographischen Nachschlagewerks stehen Antwortmöglichkeiten zur Verfügung, die automatisiert auf bestimmte Fragen ausgegeben werden können. Damit ist die Funktionalität eines Chatbots (s.u.) grob umrissen, wobei der Chatbot als maschineller Chatpartner i.d.R. noch mit einer Identität versehen wird um eine größtmögliche Nähe zum bereits durch die zwischen-liegende massive Technik distanzierten Benutzer (Stein, 2005) zu schaffen. Namensgebung und auch bildliche Erscheinung sind dabei wichtige Elemente und tragen nicht unwesentlich zum Erfolg des Bots bei.

2 Chatbot

Chatbots (= *Chat Roboter*, auch *Chatterbot* oder *Lingubots* genannt; benachbart: *Avatar*) sind Softwareagenten, die über ein Interface durch eine Kommunikation in natürlicher Sprache den Zugriff auf eine Wissensbasis bieten. Im Gegensatz zu Suchmaschinen verfügen Sie über eine i.d.R. graphisch dargestellte Persönlichkeit und einen hinterliegenden Charakter (Trogemann, 2003, S. 282ff.) mit außersprachlichen Ausdrucksformen (Mimik, Gestik, z.T. auch Stimmführung). Durch einen nur teilweise vorhersehbaren Verlauf des Diskurses und eine mögliche Lernfähigkeit geht im Gegensatz zu früheren Dialogsystemen (Nake, 1988, S. 27ff.) ein Teil der Autonomie über den Dialog von dem System und dem Entwickler des Bots an den Benutzer über. Es ist dabei umstritten, ob der Dialog mit der Maschine in Anlehnung an zwischenmenschliche Kommunikation überhaupt diesen Begriff der Kommunikation verdient oder lediglich eine Form der Interaktion darstellt (Cyranek, 1988).

Die Wissensbasis eines Chatbots kann idR. durch die Kommunikation mit den Benutzern und eventuell zusätzlichem redaktionellen Eingriff erweitert werden, d.h. der Chatbot lernt oder wird trainiert. Der Chatbot hat dabei im Verhältnis zu anderen Suchwerkzeugen durch Kommunikationsfähigkeit und Vermenschlichung eine hohe Grundakzeptanz bei geübten und ungeübten Benutzern (Wirth, 2003, S. 124f.).

Chatbots sind nicht zuletzt in ihrer moralischen, ethischen und rechtlichen Konsequenz umstritten. So wird teilweise belegt, dass die anthropomorphe Metapher durch diese starke Personalisierung eines Maschineninterfaces beim Benutzer zu hohe Erwartungen in der Kommunikation/Interaktion mit der Maschine hervorruft (Lindner, 2003, S. 9ff.). Es wird konstatiert, die Einführung menschlicher Züge in das Maschineninterface sei für Kommunikationszweck schädlich (Braun, 2000, S. 83 ff.). Zudem sei es dem Benutzer durch die Verwendung eines Chatbot nicht möglich, die Reaktion der Maschine vorherzusagen oder mehrfach erneut exakt abzurufen, worin ein erhebliches Usabilityproblem besteht (Shneidermann, 1997).

Die Historie der Chatbots geht bis in die 60er Jahre zurück, in denen ELIZA als erster bekannter Chatbot Fragen durch Rückfragen und vorgefertigte Antworten beantworten konnte (Weizenbaum, 1966). Chatbots wird neben einer Vereinfachung der Informationssuche auch ein Potential für Einsparungen in Unternehmen zugeschrieben (Braun, 2000, S. 36ff.). Schließlich bieten die Chatbots auch im Bereich des Customer Relationship Managements unternehmensrelevante Informationen durch eine verbesserte Analysemöglichkeit von Informationsbedarfen und Suchpfaden (Morphy, 2001; von Wendt, 2003, S. 46ff.). Trotz aller Begeisterung für Chatbots und ihre Wertigkeit in der Mensch-Maschine-Kommunikation (Wolff, 2003) werden sie nicht nur positiv gewertet und haben z.B. in dem Einsatz als unterstützende Lernagenten im Fremdsprachenerwerb nur wenig Erfolg gezeigt (Jia, 2007) – ganz im Gegensatz zur Chatkommunikation Mensch-Mensch in diesem Bereich (Darhower, 2002).

Der Chatbot dient in vielen bekannten Formen in der Regel der Lösung konkreter Wissens- oder Produktanfragen und damit einer spezifischen Chatsituation, die sich z.B. vom Alltagsplaudern oder Chatflirt deutlich in Kommunikationsform (Informationsbedarf) und Kommunikationsmittel (Denotation und Konnotation) unterscheidet. Zweifellos ist allerdings in der Kommunikation mit dem Chatbot mit einer im Chat generell bekannten informellen Kommunikationsform (Harnoncourt et al., 2005, S. 165ff.) zu rechnen. Die Regelhaftigkeit von themenbezogenen Chats (Döring and Pöschl, 2005, S. 154ff.) ist hier nicht anzusetzen, da der Austausch alleine zwischen Benutzer und Bot erfolgt.

Bei der Nutzung eines Chatbot wird prinzipiell in der 1:1-Kommunikation gearbeitet. Damit besteht eine Kommunikation immer nur zwischen dem Chatbot und einem Benutzer. So befinden sich im Chat-Raum als traditionellem Merkmal des Chats (Crystal, 2006, S. 11ff.) immer nur diese beiden Akteure. Technisch antwortet der Chatbot selbstverständlich auf viele Benutzer, aus der Sicht der Kommunikation relevant ist jedoch die Tatsache, dass das Gespräch Benutzer-Chatbot geschlossen ist und keine weiteren Benutzer (Mensch oder Maschine) daran teilnehmen. Damit stehen – realisiert durch entsprechende Prozesse auf dem Server beliebig viele Chatbots zur Verfügung, die eine weitgehend gleiche Funktionalität (Unterschiede in den Sitzungen mit dem Chatbot ergeben sich durch eventuelles interaktives Lernen des Chatbots, s.u.) und das exakt gleiche Aussehen haben. Die Bezeichnung als eine 1:1 Kommunikation ist damit gerechtfertigt und die fatale Problematik z.B. paralleler Chatbeiträge (Holmer and Wessner, 2005, S. 183ff.) entfällt. Insofern ist der Chatbot ein Chatsystem, und damit eine konkrete Ausprägung und Anwendung von Chat-Technologie (hierzu: Beißwenger 2005, S. 68ff.).

Gängige Steuerungs- und Kommunikationsprobleme des Chat wie z.B. mangelnde Zuordnung von Personen und deren Aussagen bei steigender Zahl der Roombenutzer (Chen and Sun, 2007) entfallen beim Chatbot. In sofern ist auch ein threaded Chat (Cadiz et al., 2000) in diesem Kontext nicht erforderlich. Chatbots wurden gemäß einer Studie unter deutschen Internetnutzern bereits 2001 als virtuelle Berater von 40 Prozent der Nutzer gewünscht (Braun, 2000, S. 13). In diesem Sinne hat z.B. die Staats- und Universitätsbibliothek Hamburg mit „Stella“ eine virtuelle Ansprechpartnerin geschaffen (Staats- und Universitätsbibliothek, 2005), die zudem für Ausführung, Wirkungsweise und Benutzung ausgezeichnet wurde (BUB, 2007). Bekannte Anwender von Chatbots sind aber auch Coca-Cola, Ford, Schwäbisch-Hall, Deutsche Bank, Bertelsmann, YelloStrom und viele andere mehr.

3 Chatbot-Technologie

Grundlagen des Chat ist das System IRC, Internet Relay Chat, das Ende der 80er Jahre bekannt wurde. Chaträume werden idR. hier von mehreren Benutzer genutzt und ein Automat (Chatbot) ist nicht vorhanden. 1:1 Sitzungen sind möglich, immer ist aber eine gesonderte Software mit eigenem Protokoll erforderlich (IRC-Client). Später in der Entwicklung anzusiedeln ist das web- und browserbasierte Chat, das in seiner Geschwindigkeit langsamer sein kann, prinzipiell aber ähnliche Funktionalitäten bietet. Ein Chatbot arbeitet prinzipiell mit eben dieser Technologie indem jeweils abwechselnd vom menschlichen Benutzer und Chatbot geantwortet wird. Denkbar und diskutiert neben den hierbei verwendeten klassischen Protokollen sind auch Streamingprotokolle (Vronay et al., 2004). In der konkreten Ausgestaltung eines Chatbots lassen sich nach der Komplexität der Reaktion des Bots verschiedene Chatbottypen in Anlehnung an und Erweiterung zu *Braun* unterscheiden (Braun, 2000, S. 43ff.):

- Der Chatbot beantwortet nur Fragen, die vollständige oder beinahe vollständige Übereinstimmung mit im System hinterlegten Fragen haben; eine Standardantwort wird gegeben.

- Der Chatbot antwortet auf eng definierte Fragetypen mit einer gewissen, begrenzten Beliebigkeit innerhalb der nachgefragten Information.

- Der Chatbot arbeitet mit Mustererkennung als einer zentralen Technologie (von Wendt, 2003) in der Fragestellung und antwortetet dann mit im System hinterlegten Antworten. Dieses *Cased-Based Reasoning* oder *Nearest-Neighbor Classification* definiert Alice (zu Alice s.u.) „For every input, we find the best matching "case" in the pattern set, and generate a reply based on the associated template" (Artificial Intelligence Foundation, 2007b). Der Chatbot antwortet auf eine ihm unbekannte Frage mit einer Rückfrage und sieht die dann erhaltene Rückantwort des Benutzers als mögliche Beantwortung der vom ihm als Bot gestellten Rückfrage an. Ein Frage-Antwortpaar wurde gebildet. Die Gefahr von Falschinformation des Bots ist hier aber nicht unerheblich, denn der Bot muss – wenn kein redaktioneller Eingriff vorhanden ist – alle Aussagen als wahr annehmen.

- Der Chatbot hat verschiedene Verhaltensweisen (= Programme) für verschiedene Kommunikationsformen und Themen, die je nach initialer Eingabe durch den Benutzer genutzt werden und thementypische Fragestellungen beantworten können.

Zahlreich genutzt als Chatbot ist die Technologie, auf der Alice aufbaut (Artificial Intelligence Foundation, 2007a; Reichle, 2006). Alice basiert auf einer definierten XML-Sprache für eine Mustererkennung von Fragen an den Bot und deren Antworten, AIML (Wallace, 2002). Durch die Trennung von Interpretation und Darstellung einerseits und Wissensbasis in AIML andererseits ist der Aufbau der Wissensbasis softwareunabhängig möglich. Die Interpretation und Darstellung können durch frei verfügbare Programme in verschiedenen Sprachen durchgeführt werden. Verschiedene Chatbots auf Basis von Alice und AIML sind belegt (Braun, 2000, S. 44ff.) und haben dabei u.U. auch eine gewisse Lernfähigkeit. Als eine vergleichbare Möglichkeit kann bei Pandorabots auf der Grundlage von Alice ein Bot generiert werden, der dann von dem dortigen Server gehostet wird (Boguschewski, 2005).

4 Chatbot für ein Online-Lexikon

Ein Chat eignet sich prinzipiell auch für Informations- und Beratungsgespräche (Beißwenger, 2005, S. 10), wobei die automatisierte Anwendung als Chatbot noch kaum beschrieben ist, jedoch auf allgemeine Eigenschaften kommunikativer Schnittstellen in der Informatik zurückgeführt werden kann (Schmitt, 1983, S. 22 ff.). Es gilt dabei einen möglichst hohen Grad an Interaktivität (Nake, 1988, S. 16 f.) im Dialog zwischen Nutzer und lexikographischem Nachschlagewerk zu erzielen. Dabei wird davon ausgegangen, dass sich das Diskursmuster im wesentlichen auf den Verlauf *Frage* → *Antwort* mit eventuell *n* zwischengeschalteten Rückfragen und Rückantworten reduzieren lässt. Die Beratungsfunktion einer Verbraucherzentrale eignet sich damit für einen konkreten Test mit einem Chatbot. Ausgangspunkt sei hierbei ein Online-Lexikon für junge Verbraucher, das an der generellen Beratungs- und Informationsarbeit einer Verbraucherzentrale ansetzt (Geeb, 2006; Geeb and Spree, 2005). Dieser Rahmen determiniert bereits die Domäne durch die im Lexikon vorhandenen Themen, die sich ausschließlich auf die generelle Beratungsthematik der Verbraucherzentrale bezieht. Die Wissensbasis des Chatbots lässt sich damit zumindest domänenspezifisch als Zielsetzung aus diesem fachlexikographischen Nachschlagewerk ableiten. Für den Chatbot steht dabei in Analogie zu einem korpusbasierten Training des Bots (Abu Shawar and Atwell, 2005) eine automatisierte Wissensbasis zur Verfügung, die allerdings im Gegensatz zu einem allgemeinsprachlichen Korpus als lexikographisches Nachschlagewerk eine differenzierte fachliche Informationstruktur aufweist.

Aus syntaktischer und semantischer Sicht wäre der Umgang mit besonderen Ausdrucksformen, die im Chatbereich außerhalb von Bots sonst bekannt sind wie Emoticons und Abkürzungen (Diekmannshenke, 2005) im Chatbot theoretisch realisierbar. Eine große Gruppe der Chatbenutzer ist regelmäßiger Nutzer dieser Emoticons (Husmann, 1998, S. 74). Andererseits ist dieser umfangreiche Gruppencode (Husmann, 1998, S. 34ff.) dem nicht unbedeutenden Anteil der anzusetzenden Chatbotnutzer, der die Chatgepflogenheiten des Internets nicht kennt (Medienpädagogischer Forschungsverbund Südwest, 2007a), unbekannt und wäre für diese Gruppe in den Antworttexten des Chatbots damit ein Kommunikationshindernis. Eine Realisierung der Emoticons in Frage und Antwort würde außerdem eine intensive Pflege eines entsprechenden Datenbestandes erfordern. Zudem ist der Mehrwert in der konkreten Benutzerintention „Sachinformation“ (Geeb, 1998, S. 58ff.) nicht wirklich erkennbar, denn diese Chatausdrucksformen dienen insbesondere der Sozialisierung und Abgrenzung von Gruppen (Crystal, 2006, S. 171ff.), nicht aber dem Informationsbedarf, der durch den Chatbot abgedeckt werden soll. Dies gilt um so mehr, als ein wesentlicher Teil der Zielgruppe das Medium Internet zur Zeit (Shuli and Nielsen, 2002, 110 f.) in der Hauptsache für Informationszwecke nutzt (Medienpädagogischer Forschungsverbund Südwest, 2007b, 44 f.). Die Emoticons sollten damit im Sinne der *substitution normalized form* wie aus Alice bekannt ersatzlos aus der Eingabe gelöscht werden (Abu Shawar and Atwell, 2002).

Trotzdem muss aber mit einer informellen Ausdrucksweise und auch mit dem Verzicht auf gewissen grammatikalische Regeln wie z.B. Groß-/Kleinschreibung gerechnet werden (Crystal, 2006, S. 176f.; Puck and Exter, 2005, S. 297f.), und sogar Tippfehler müssen in Kauf genommen und bei der der Anwendung des Gesprächs auf die (lexikographische) Wissensbasis bedacht werden (Willand, 2002, S. 59).

Kohärenzbildung in computervermittelter Kommunikation durch Training / Schulung (Zumbach and Spraul, 2005, S. 380) ist in diesem Fall kaum durchführbar, da die Benutzer ständig wechseln und eine vollständig heterogene und unbekannte Gruppe bilden. Auch scheint die Benutzung eines Chatbots prinzipiell auf Grund des natürlichsprachlichen Interfaces keine weiteren Erläuterung zu erfordern (Wirth, 2003, S. 124f.). Tatsächlich ist Chatten mit seiner Chatiquette ein relativ etabliertes Kommunikationssystem, das aber nicht in seinem vollständigen Umfang in einem Chatbot realisiert wird – und diese Grenzen und Unterschiede zum normalen Chat müssen im Sinne der Benutzerfreundlichkeit vermittelt werden. Ein entsprechendes Training kann und muss sich in diesem Fall auf eine Kurzanweisung beschränken. Diese Kurzanweisung kann in Textform gegeben werden, sollte aber zur Erhöhung der Akzeptanz und Beachtung auch als Kurzvideo (z.B. Flash) mit menschlicher Stimme dargeboten werden oder multimedial von der Chatfigur und damit dem Chatbot selbst geäußert werden. Geschriebene Anweisungen und Hilfstexte werden in der Regel nur wenig beachtet, insbesondere im Internet und bei einer jüngeren Zielgruppe wird eine intuitive Benutzung des Systems erwartet.

Durch die Kommunikation mit dem Computer, der im vorliegenden Konzept in relativ natürlicher Sprache antworten soll, werden voraussichtlich zahlreiche Nutzer eine gewisse emotionale Bindung mit diesem virtuellen Gesprächspartner entwickeln (Reeves and Nass, 1996). Somit kommt der graphischen und evtl. akustischen Ausgestaltung der Chatbotfigur eine wichtige Bedeutung im Sinne einer ästhetischen Dimension von Information zu (Hentschläger and Wiener, 2002). Die Mehrzahl der aktiven Chatbots ist weiblich (auch männliche sind belegbar: z.B. Forstliche Versuchs- und Forschungsanstalt Baden-Württemberg (2007)) und verfügt über ein wissenschaftlich sicher nicht belegbares aber als Grundkonsens doch bekanntes Aussehen, das dem gängigen Schönheitsideal (räumlich und zeitlich) entspricht. Bei der Stimmführung weiblicher Chatbotcharaktere hat sich offenbar eine tiefere Frauenstimme gegen eine höhere, femininere Stimme als erfolgreicher durchgesetzt (Braun, 2000, S. 63ff.). Im Hinblick auf die Zielgruppe ist eine junge erwachsene Person als Chatbotgrafik zu bevorzugen – so werden Jugendlichkeit und Erwachsenenkompetenz in dem Beratungsgespräch verbunden, wobei konzeptionell im Sinne der Gleichstellung und Gleichberechtigung dem Benutzer die Wahl zwischen einer männlichen und einer weiblichen Chatbotfigur gegeben werden sollte. Die Figur darf dabei nicht statisch sein, sondern muss möglichst synchron mit dem Gesprächsverlauf Mimik und Gestik zeigen und ändern (Braun, 2000). Diese Gestik und Mimik muss dabei im unten genannten Modell der Antwortgenerierung immer rudimentär bleiben, da eine syntaktische und satzsemantische Analyse nicht vorgesehen ist. Außersprachliche Ausdrucksformen zeigen damit im vorliegenden Entwurf nur eine Grundstimmung der Kommunikation wie dies auch vielfach belegt und in der zugrundeliegenden Technologie begründet ist – mit Alice (Artificial Intelligence Foundation, 2007a) als prominentestem Vertreter.

Die Bindung des Nutzers an den Chatbot darf allerdings nicht zur Bildung neuer Identitäten führen, wie dies in normalen Chats teilweise möglich ist. Dort können – je nach System – Chatnutzer durch Figuren (Avatare) und ihr Chatverhalten teilweise oder völlig neue Identitäten im Verhältnis zum wirklichen Leben annehmen (Filinski, 1998, S. 107ff.; Willand, 2002, S. 79ff.). Diese Verhaltensweise wäre im Rahmen des Chatbot technisch realisierbar, aber sehr aufwendig und zudem nicht produktiv. Diese Identitätsbildung auf Nutzerseite ist letztlich nicht Teil eines reinen Informationsbedürfnisses und würde durch die vielfachen sozialen und psychologischen Elemente den Chatbot inhaltlich in seiner Funktion als Informationsassistent mit der vorhandenen Wissensbasis des Lexikons überfordern.

Der zu konzipierende Chatbot hat keinen Moderator, d.h. des handelt sich einerseits nicht um einen moderierten Chat und andererseits werden nicht beantwortbare oder beantwortete Fragen der Nutzer nicht automatisch an einen HelpDesk o.ä. weitergeleitet, der zeitnah antworten kann. Dieses Verfahren der zeitnahen Weiterleitung unbeantworteter Fragen an den Chatbot (z.B. produktbezogene Themen) ist aus der Wirtschaft bekannt und wünschenswert (Salimi, 2003, S. 136ff.; Thommes, 2001), für einen öffentlichen Träger ohne Nutzerfinanzierung oder direkte Kopplung eines Customer Relationship Managements an einen wirtschaftlichen Erfolg aber kaum durchführbar.

Die Lernfähigkeit des Chatbots ist erwünscht um die Akzeptanz des Chatbots bei den Nutzern zu fördern. Die Erstellung eines zu veröffentlichenden Transkriptes des Chat, das bei moderierten Chats u.a. in der Politik bekannt ist, ist an dieser Stelle nicht sinnvoll, da ein Nacharbeiten der Suchanfragen und Suchstrategien einzelner Nutzer durch andere Nutzer zeitaufwendig wäre. Dieser Zeitaufwand entspricht aber eben nicht dem Konzept der schnellen und benutzergesteuerten Information eines Chatbots. Andererseits ist der Chatverlauf in einem systeminternen Protokoll zu speichern, das dann für die Überprüfung der Funktionsfähigkeit, Nutzbarkeit und Anwendungsfelder durch redaktionelle Administratoren im Überblicksverfahren genutzt werden kann.

Zu beachten sind bei der Erstellung eines Chatbots in jedem Fall Kriterien der Benutzbarkeit von Webseiten und deren Dienste allgemein (z.B. Nielsen 2001). Hier ist zum einen eine Heuristik (Nielsen, 1993, S. 115ff.) für die Evaluation des Chatbots und seiner Charaktere zu entwickeln; zum anderen sind exemplarische Labortests mit einer Videokontrolle der Benutzerarbeit mit dem Chatbot anzufertigen (Nielsen, 1993, S. 200ff.). Insbesondere die Wartezeit des Benutzers auf eine Antwort ist bekannter Maßen ein wesentlicher Erfolgs- und Usabilityfaktor (Wirth, 2003, S. 125) und es gilt, diese durch eine entsprechende Konzeption des Programmflusses sowie durch die Programmierung und Gestaltung der Wissensbasis möglichst gering zu halten.

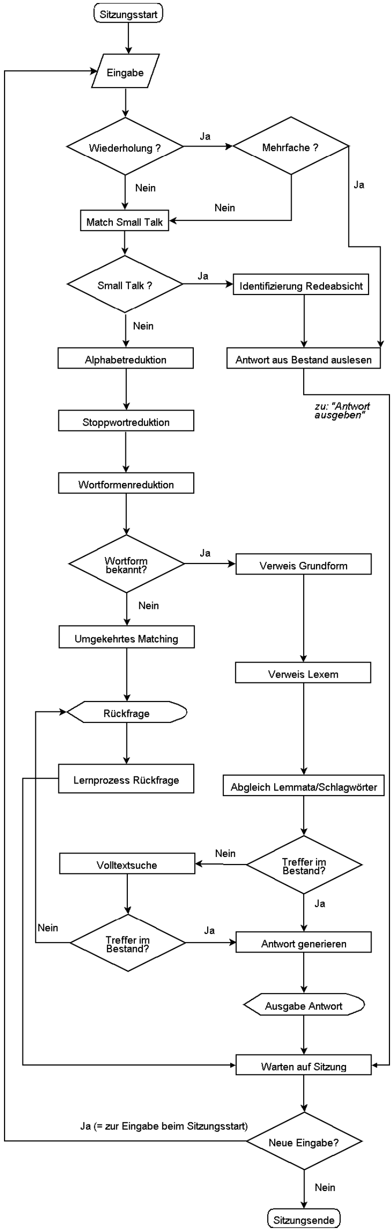


Abbildung 1: Schematischer Programmablauf eines Chatbots

Auch die Grundsätze der Barrierefreiheit der *Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz* (Behindertengleichstellungsgesetz, 2002) sind für fachlexikographische / terminologische Produkte aus ethischen Gründen – und im vorliegenden Fall auch z.T. aus rechtlichen Gründen – zu beachten, und so muss der Chatbot in reiner Textform steuerbar und verständlich sein. Auch muss aus diesem Grund nebenläufig zur Videokurzeinführung des Systems (s.o.) ein Textstream mit diesen Informationen angeboten werden. Der Programmablauf des Chatbots lässt sich wie in Abb. 1 dargestellt zusammenfassen.

Der Chatbot steht prinzipiell in unbegrenzt vielen Ausgaben zur Verfügung. Jede Kommunikation mit einem Benutzer von deren Beginn zu deren Ende wird als Sitzung bezeichnet und technisch gesehen auch als solche realisiert. Der Sitzungsbeginn entspricht der ersten Eingabe des Benutzers. Die Sitzungslänge und damit die Entscheidung, wann eine neue Sitzung beginnt und damit das Gedächtnis des Chatbots im Bezug auf diesen Benutzer gelöscht wird, muss getroffen werden, lässt sich aber jederzeit problemlos revidieren. Ein vorstellbares Vorgehen wäre nach 60 Sekunden Inaktivität die Sitzung serverseitig durch Löschen der Sitzungsdaten (= Gedächtnis des Chatbots) zu beenden. Ein Kurzzeitgedächtnis des Chatbots ist aber in jedem Fall erforderlich (Boguschewski, 2005).

Bei der ersten Eingabe tritt die darauf folgende Verzweigung mit Prüfung auf *Wiederholung* ist bei Sitzungsstart nicht aktuell. Die Eingabe wird daraufhin durch Patternmatching auf *Small-Talk* untersucht wie dies durch Alice (Artificial Intelligence Foundation, 2007a) und z.B. ein deutsches Small-Talk-AIML-Set zu diesem Zweck belegt ist (Drossmann, 2001). Als Beispiel und Auszug aus dem genannten Set hier die übliche Frage nach der Identität des Bot, die dann noch in verschiedenen Schreibvarianten vorgelegt wird:

```
...
<category>
        <pattern>BIST DU EIN MENSCH</pattern>
        <template>Nein, ich bin eine Maschine.</template>
</category>
<category>
        <pattern>BIST DU EIN PROGRAMM</pattern>
        <template>Ja, ich bin ein Programm.</template>
</category>
...
<category>
        <pattern>DAS DENKE ICH *</pattern>
        <template>Aber Du bist Dir nicht sicher, oder?</template>
</category>
...
```

Alle Gesprächsversuche mit dem Chatbot, die nicht dem reinen Informationsbedürfnis dienen, werden hier durch hinterlegte Muster vergleichbar diesen rudimentären Beispielen aus AIML geprüft, wobei die Wirkung einer Erweiterung von AIML durch reguläre Ausdrücke und Stemming überprüft werden muss. Dies betrifft Fragen mit Inhalten wie *„Wie alt bist Du“*, *„Wer hat Dich programmiert“* oder im Stil von *„Willst Du*

Dich mit mir treffen". Diese Art der Kommunikation mit dem Chatbot ist durchaus üblich (Davidson, 2005; Reichle, 2006). Wenn es sich um Small-Talk handelt, wird dieser weiter auf die *Redeabsicht* untersucht und mit einer passenden Antwort aus dem Antwortbestand assoziiert. Handelt es sich um ernst gemeinte aber nicht relevante Fragen (*„Wie alt bist Du"* etc.) werden die Antworten ähnlich ernst gemeint sein, aber freundlich auf den eigentlichen Sinn des Chatbots hinweisen. Enthalten die Fragen Zweideutigkeiten, werden diese zurückgewiesen (Lindner 2003, 11; beim Chatbot *Stella*: Bachfeld et al. 2005, S. 209) und ggf. mit einem neutralen Satz beendet. Im Gegensatz zu einem normalen Chat (Willand, 2002, S. 64f.) kann der Bot den unerwünschten Nutzer auf Grund dessen völligen Anonymität aber nicht technisch ausblenden und muss immer wieder mit entsprechenden Eingaben rechnen. Es liegt in der Natur der Sache, dass nicht alle Small-Talk Fragen vorhergesehen werden können und deswegen u.U. weitere Prozesse der normalen Beantwortung trotz Small-Talk eingeleitet werden. Als Alternative könnte man die Definition des Small-Talk durch ein Patternmatching gegen die bekannten Informationsfragen definieren, würde dann aber einen Großteil der unbekannten aber relevanten Anfragen als Small-Talk deklassifizieren. Zudem wäre die Lernmöglichkeit des Chatbots, die an späterer Stelle in der Bearbeitung normaler Anfragen vorliegt, wesentlich eingeschränkt. Bei der sprachlichen Analyse der Eingaben wäre eine prinzipielle Unterscheidung zwischen negativer und positiver Grundaussage hilfreich zur Steuerung des weiteren Gesprächsverlaufs (Turney, 2003). Dieser Analyseschritt wird aber auf Grund des hierfür erforderlichen umfangreichen Textkorpus für dieses Vorhaben zunächst nicht in die Konzeption aufgenommen.

Wenn normale Anfragen vorliegen, werden diese zunächst durch eine *Alphabetreduktion* geführt und nicht etwa mit einem extremen Patternmatching (von Wendt, 2003, S. 42ff.) bereits eine Antwort generiert. Bei der Alphabetreduktion werden alle nichtalphabetischen Zeichen entfernt mit Ausnahme von diakritischen Zeichen und Ligaturen sowie Bindestrichen, die ohne Zwischenraum zwischen zwei Buchstabenblöcken vorkommen. Dies könnten Komposita sein, die die Benutzer auf Grund der vermutlich weniger regeltreuen Schreibweise in Anlehnung an andere Sprachen mit Bindestrich schreiben. Zahlen (arabisch oder römisch) könnten ebenfalls für die Semantik der Anfrage interessant sein; daher wäre dieser Teil des Programmablauf so zu konstruieren, dass die Liste der gültigen Zeichen zu jedem Zeitpunkt einfach (z.B. ohne weiteres Kompilieren von Quellcode) erweitert werden kann; im genannten Fall der Zahlen wäre die Wirkung mit oder ohne Zahlen dann zu testen. Der Abschluss einer Benutzereingabe durch ein Fragezeichen wird in der Sitzung als besondere Information gespeichert; in diesem Fall kann davon ausgegangen werden, dass definitiv eine Frage gestellt wurde. Das Fragezeichen an sich wird aber aus der Eingabe gelöscht.

Gängige Wörter, die sehr häufig auftreten und im Retrieval von Informationen daher eine geringere Bedeutung für die Erfassung des Inhalts der Benutzereingabe haben, werden im nächsten Schritt durch eine *Stoppwortreduktion* entfernt; nicht gelöscht werden dürfen Wörter wie *„Ja"* und *„Nein"* etc., die bei Rückfragen Aufschluss über Sachzusammenhänge geben. Diese Stoppwortliste muss ebenfalls erweiterbar sein (z.B. Datenbank als Basis) und lässt sich durch einen entsprechenden Textkorpus z.B. auf der Grundlage des Lexikons, für den der Chatbot geschrieben wurde, gewinnen und verfeinern. –Schließlich erfolgt eine *Wortformenreduktion*, in der die eingegeben Buchstabenblöcke durch Patternmatching auf eine Grundform, das Lexem, zurückgeführt werden. Die Datengrundlage für diesen Arbeitsschritt ist sehr aufwendig – und je aufwendiger sie gehalten wird, je besser wird das Ergebnis dieses Schritts sein. So müssen prinzipiell alle Wortformen der Lexeme, die Bestand des Lexikons als Lemmata oder auch in beschreibenden Texten sind und die nicht zur Stoppwortliste gehören sollen, hier vermerkt werden. Zusätzlich sind noch umgangssprachliche und gruppensprachlich relevante Schreibvarianten zu berücksichtigen. Alle Formen und Varianten zeigen über ihre Grundform auf ein Lexem (*Brille, Brillen, Nasenfahrrads, ... → Brille*). Somit wird das Thema der Suchanfrage eingegrenzt. Bei Suchanfragen wie *„brauche ich eine bildschirmbrille?"* wird damit möglicherweise keine Wortform gefunden und bei der Verzweigung *Wortform bekannt?* wird der Programmfluss auf eine Rückfrage umgelenkt. Die Rückfrage des Chatbots sucht durch den Schritt *Umgekehrtes Matching* alle Lexeme heraus, die in der unbekannten Wortform enthalten sind wie z.B. *Bildschirm* und *Brille*. Sollten sich daraus – wie im vorliegenden Fall – Komposita bilden lassen, die der nicht gefundenen Wortform ganz oder teilweise entsprechen, ist die Rückfrage mit Rückgriff auf die beiden bekannten Wortformen einfach: *„Bildschirmbrille kenne ich nicht. Meinst Du Bildschirm und Brille?"*

Hier wurde bereits über die Sitzung der Lernprozess gestartet, d.h. die initiale Frage des Benutzers wird gespeichert und auch die Antwort des Chatbots (= Rückfrage) wird gespeichert usw. Nach Beendigung der Sitzung werden diese Daten strukturiert zur redaktionellen Nachbearbeitung (= eventuelles Einpflegen von *Bildschirmbrille*) abgelegt. Nun wird auf eine neuerliche Eingabe des Benutzers gewartet. Wenn diese nach binnen der eingestellten Frist erscheint, wird die Sitzung fortgeführt und die neue Eingabe jetzt auf eine Wiederholung der ursprünglichen Benutzereingabe untersucht. Wird die ursprüngliche Eingabe mehrfach wiederholt (auch dieser Wert muss konfigurierbar sein), wird eine entsprechende Antwort aus dem festen Antwortbestand ausgelesen und der Timeoutzähler der Sitzung gestartet. Mit anderen Worten kann der Benutzer *n*-fach denselben Text eingeben und wird eine entsprechende Antwort im Stil von *„Das ist mir nicht neu"* erhalten, wird dem Chatbot aber keine weitere Aktion entlocken können bzw. vermutlich durch den Timeout gestoppt. Der Timeout ist technisch gesehen nicht erforderlich, da der Bot durch die gleichbleibende Antwort nicht in seiner Funktion eingeschränkt wird. Da jedoch eine sinnvolle Kommunikation mit dem Bot gefördert werden soll, wäre der Timeout ein entsprechendes Mittel hier einzugreifen.

Wird die ursprüngliche Rückfrage des Chatbots mit *„Ja"* beantwortet, ist einerseits der Lernprozess abgeschlossen (s.o.) und andererseits werden die beiden zuvor genannten Lexeme als relevant eingestuft. In diesem Fall – so wie auch wenn die Wortform bereits bei der ersten Anfrage auf ein Lexem reduziert werden konnte – wird über den Schritt *Verweis Grundform* die Grundform gesucht, die dann in *Verweis Lexem* auf ein bestimmtes Lexem zeigt und in der Datenbasis auch Informationen über das Verhältnis der Grundform zum Lexem enthält. So kann hier z.B. vermerkt werden, dass *Nasenfahrrad* Jugendslang für *Brille* ist. Durch diese Kennzeichnung von Bedeutungsbeziehungen (Geeb, 1998, S. 164ff.) wird die Pflege der Datenbasis erheblich erleichtert, indem Bereiche wie Slang oder Abstraktionsbeziehungen kontrolliert gesichtet werden können.

Über die Kenntnis des Lexems kann der Chatbot dann im *Abgleich Lemmata/ Schlagwörter* die Lexikonartikel in der Sitzung ablegen, die über das Lexem referenziert sind. Das sind auf einer ersten Stufe Lexikonartikel, in denen das Lexem als Lemma auftritt. Wenn diese Suche nicht zum Erfolg führt erscheinen in einer zweiten Stufe alle Lexikonartikel relevant, in denen das Lexem als Schlagwort genannt ist, wobei eine intellektuelle Verschlagwortung der Lexikonartikel vorausgesetzt wird. Wird dann durch die Verzweigung *Treffer im Bestand?* ein passender Artikel gefunden, sind die folgenden Schritte *Antwort generieren* und *Ausgabe Antwort* problemlos. Wird aber unter *Treffer im Bestand?* kein Match gefunden, muss als dritte und damit relativ unscharfe Suche eine Suche des gefundenen Lemmas im Volltext des Lexikons gestartet werden. Führt dies über das folgende *Treffer im Bestand?* zu einem Ergebnis, kann eine Antwort angeboten werden, die aber semantisch an diese unscharfe Suche angepasst werden muss. Konnte kein Match gefunden werden, wird die *Rückfrage* angestoßen, die dann vom Benutzer eine Neuformulierung der Eingabe erbittet. Nach Ausgabe der Antwort als Link auf das entsprechende Lemma mit einsätziger kurzen Vorformulierung der semantischen Informationen/ Definition (Geeb, 1998, S. 145ff.) wird dann über den Timeout auf eine Neueingabe des Benutzers gewartet oder die Sitzung dementsprechend geschlossen.

Das gezeigte Modell führt allerdings in definierbaren Fällen, die bereits im jetzigen Konzeptstadium vorhersehbar sind, zu einer problematischen Vorgehensweise. So wird u.U. nur ein Lexem über das genannte Verfahren aus der Wissensbasis gefunden und als Antwort generiert; tatsächlich waren aber mehrere Lexeme relevant – aber dem System nicht bekannt. Als Beispiel sei hier genannt: *Ist Rippen von CDs verboten* wobei *CD* als Lexem in der Wissensbasis bekannt und damit antwortfähig ist während *Rippen* als Slang für *Auf den Rechner kopieren* nicht erkannt wurde. Die Antwort wäre damit allgemein zu Compactdiscs, nicht aber zur Frage der Legalität des Kopiervorgangs. Diese Problematik lässt sich in der gezeigten Annahme nur lösen, indem der Benutzer über eine Rückfrage erneut eindeutig nach *Rippen* nachfragt und damit einen protokollierten Lernprozess ausführt.

Ebenfalls problematisch ist das Auffinden meherer Lexeme gleicher Relevanz, d.h. durch die Reduktion werden mehrere Lexeme lokalisiert, die alle als Lemma vorhanden sind und damit höchste Relevanz besitzen. So fragt der Benutzer *Machen Alkopops süchtig* und erhält aus der Wissensbasis die beiden lemmatisierten Lexeme *Alkopops* und *Sucht*. Als Lösung kann ihm nur eine differenzierte Antwort angeboten werden die besagt, dass mehrere relevante Informationen vorliegen.

Eine ähnliche Konstruktion ergibt sich durch Homonymie und Polysemie, die letztlich nur durch eine Fachsystematik in Verbindung mit den Lemmata aufgelöst werden könnte. Diese Fachsystematik muss dann wieder auf die Lexeme nach dem Verweis aus den reduzierten Grundformen abgebildet werden und bei mehreren gefundenen relevanten Lexemen müsste ein Vergleich dieser damit gefundenen fachlichen Einträge und ggf. ein Entscheidungsprozess zur Relevanz eingeleitet werden. Dieses Verfahren ist einerseits umfangreich im Programmfluss aber andererseits vor allen Dingen aufwendig in der redaktionellen Arbeit und wurde daher hier nicht berücksichtigt.

5 Ausblick

Die in der Literatur anzutreffende Differenzierung zwischen Fachlexikographie und Terminologie (z.B. Budin and Wright, 1997; Arntz et al., 2004; Bergenholtz, 1995; Wiegand, 1979) wurde hier übergangen und kann nicht Thema aus der Sicht des Chatbotkonzepts sein. Der Ansatz ist prinzipiell für beide Bereiche vergleichbar, denn die für den Chatbot erforderliche Wissensbasis wird auf ähnlicher Grundlage generiert. Durch die zahlreichen Nutzungsmöglichkeiten und Nutzungsänderungen im Bereich der neuen Medien wird die Grenze zwischen Online-Fachlexikographie und Online-Terminologie zukünftig noch mehr an Prägnanz verlieren und nur in Randbereichen wie z.B. Normung einerseits und etymologischen Fachwörterbüchern andererseits noch deutlich sichtbar sein. Ein Chatbot im vorgestellten Konzept könnte von dieser Differenzierung nur rudimentär profitieren.

In der Chatkommunikation wird auf der Nutzerseite oftmals eine Verbesserung der Chat-Kompetenz gefordert (Döring and Pöschl, 2005, S. 159). Die ebenfalls in diesem Zusammenhang oftmals geforderte Verbesserung der technischen Möglichkeiten betrifft in der Regel den Wunsch nach einer Vollduplex-Kommunikation. Auch wenn diese Entwicklung möglich wäre, löst sie aber nicht das Problem des „fehlenden Augenkontakts“ und damit des Sprecherwechsels in einem Gespräch im Chat (Zumbach and Spraul 2005, S. 379; Beißwenger 2005, S. 85) und somit auch mit dem Chatbot. Nicht unähnlich einer Telefonkonferenz ist eine nonverbale Absprache über die Sprecherreihenfolge im Netzwerk nicht möglich, wenn nicht zu einer strikten Redefolge gegriffen werden soll. Diese Problematik ist zwar für den Chatbot zunächst weniger wichtig, bei vollem Funktionsumfang könnte eine Vollduplexkommunikation mit dem Chatbot jedoch die Eröffnung eines Chatbot-Rooms ermöglichen.

Eine Lösung der Halbduplexproblematik könnte sich durch die Verwendung eines factchat (Harnoncourt et al., 2005) ergeben, in dem die Beiträge von Benutzern zeitlich und räumlich (zweidimensional) frei positioniert werden können. Die Eingabe von Texten ist für alle Teilnehmer fast zeitgleich sichtbar und gleichzeitig ist eine Ausgabe oder Eingabe von Beiträgen durch die Zeitschiene auch diachron möglich. Für einen Chatbot erscheint diese Lösung zunächst aber nicht erforderlich, da in der Regel auf die 1:1 Kommunikation gesetzt wird, in der eine synchrone Gesprächsführung nur sehr bedingt erforderlich ist. Zudem hat die räumliche Kommunikation mit mehreren Benutzern in diesem Fall auf Grund der voraussichtlich kurzen Gesprächsdauer und der stets besonders nutzerbezogenen Informationsbedürfnisse nur einen Mehrwert als Betrachtung historischer Daten. Schließlich erfordert die Realisierung der Antworten des Chatbots in einem mehrdimensionalen Raum auf Grund der z.T. erforderlichen Rückfragen weitere Überlegungen und Tests.

Der Lebenszyklus eines Chatbots in der angedachten Form ist vermutlich kürzer als der Lebenszyklus eines traditionellen fachlexikographischen Nachschlagewerks, denn nicht zuletzt die Sprache und das sprachliche Verhalten werden sich im Takt der schnellen Technikentwicklung weiterentwickeln (Crystal, 2006, S. 57ff.). Am Horizont erwartet man bereits Chatbots mit Spracherkennung in eben dieser z.T. übereinfachen, wenig strukturgebundenen und fast immer informellen Sprache des Chats. Chats mit mimikgenerierenden, im 3D-Raum beweglichen Avataren sind bereits belegt und untersucht (Cassel, 2003; Drucker et al., 2000); auch das wäre eine Zukunftsperspektive für einen Chatbot, wobei das Lesen und Verstehen der Mimik und Gestik des Gegenüber (also des Nutzer-Avatars) eine besondere Herausforderung für die Entwicklung des Chatbots ist. Der Lebenszyklus eines Chatbots verlängert sich durch sein Training, d.h. durch die Pflege der Antworten auf Grund der gestellten Fragen – in inhaltlicher und sprachlicher Sicht. In ähnlicher Weise positiv verlängernd wirkt sich auf den Lebenszyklus die Erweiterung der fachlexikographischen / terminologischen Datenbasis aus.

Weitere Möglichkeiten ergeben sich durch eine Mehrsprachigkeit des Chatbots – mit all den Schwierigkeiten, die die oftmals fehlende volle Äquivalenz zwischen zwei Lexemen in L1 und L2 mit sich bringen (Svensén, 2004, S. 310ff.). Hinzu kommt der unverhältnismäßig steigende Pflegeaufwand für die Wissensbasis des Chatbots in mehreren Sprachen und evtl. mehreren Zeichensystemen.

Umfangreich in der Realisierung wäre eine syntaktische und (satz-) semantische Analyse der Benutzereingaben an den Chatbot. Sprachliche und außersprachliche Anteile der Kommunikation des Chatbots lassen sich aber auf diese Weise voraussichtlich verbessern. Der Mehrwert dieses Ansatzes für den Benutzer im Verhältnis zur Wissensbasis (fachlexikographisches Nachschlagewerk) und im Verhältnis zum Implementierungsaufwand wäre dabei noch zu bewerten.

Die Frage der konkreten technischen serverseitigen Realisierung des Chatbots und damit die Wahl von Plattform, Programmiersprache und Datenhaltung z.B. in einer relationalen oder objektorientierten Form wurde hier nicht diskutiert uns ist zunächst für die allgemeine Konzeption nicht erheblich sondern projektabhängig. Fest steht, dass die Wahl der Mittel auf der Seite des Klienten immer die Grundsätze von Usability und Barrierefreiheit beachten muss. Der naheliegende Einsatz von Technologien wie z.B. Flash oder Ajax auf der Clientseite ist daher genau in ihrer Konsequenz zu überprüfen. Der Bot muss im Betrieb insgesamt einem fortlaufenden Qualitätstest im Hinblick auf Lernfähigkeit, Wissensbasis, Kontexterkennung und Gedächtnisfunktion unterzogen werden (Vetter, 2003). Nur so lässt sich die Qualität der Chatkommunikation mit dem fachlexikographisch/terminologisch unterstützten Chatbot stetig verbessern.

Literatur

Abu Shawar, B. and Atwell, E. (2002). A comparison between Alice and Elizabeth chatbot systems. Research Report 2002.19, University of Leeds School of Computing.

Abu Shawar, B. and Atwell, E. (2005). Using corpora in machine-learning chatbot systems. *International Journal of Corpus Linguistics*, 10(4):489–516.

Arntz, R., Picht, H., and Mayer, F. (2004). *Einführung in die Terminologiearbeit*. Studien zu Sprache und Technik. Olm, Hildesheim, 5. edition.

Artificial Intelligence Foundation (2007a). The A. L. I. C. E. Artificial Intelligence Foundation. Online unter http://www.alicebot.org.

Artificial Intelligence Foundation (2007b). Gespräch mit Alice. Online unter http://www.pandorabots.com/pandora/talk?botid=f5d922d97e345aa1.

Bachfeld, S., Christensen, A., and Christof, J. (2005). Automatische Auskunftsgespräche? Anforderungen an den Einsatz von Chatbots in Bibliotheken. *Zeitschrift für Bibliothekswesen und Bibliographie*, 89:205–218.

Behindertengleichstellungsgesetz (2002). Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz. Online unter http://bundesrecht.juris.de/bundesrecht/bitv/gesamt.pdf.

Beißwenger, M. (2000). *Kommunikation in virtuellen Welten: Sprache, Text und Wirklichkeit. Eine Untersuchung zu Konzeptionalität von Kommunikationsvollzügen und zur textuellen Konstruktion der Welt in synchroner Internet-Kommunikation, exemplifiziert am Beispiel eines Webchats*. Ibidem, Stuttgart.

Beißwenger, M. (2005). Interaktionsmanagement in Chat und Diskurs. Technologiebedingte Besonderheiten bei der Aushandlung und Realisierung kommunikativer Züge in Chat-Umgebungen. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 63–87. Ibidem, Stuttgart.

Beißwenger, M. and Storrer, A. (2005). Chat-Szenarien für Beruf, Bildung und Medien. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 9–25. Ibidem, Stuttgart.

Bergenholtz, H. (1995). Wodurch unterschiedet sich Fachlexikographie von Terminographie? *Lexicographica*, 11:50–59.

Boguschewski, F. (2005). Selbstredende Homepage. Pandorabots: Wenn Software antwortet. *c't - Magazin für Computertechnik*, 12:214.

Bradner, E., Kellogg, W. A., and Erickson, T. (1999). The adoption and use of 'babble': A field study of chat in the workplace. In *Proceedings of the Sixth European conference on Computer supported cooperative work*, pages 139–157.

Braun, A. (2000). *Chatbots in der Kundenkommunikation*. Springer, Berlin/ Heidelberg/ New York.

Bremmer, C. (2005). Chats im eLearning. Rollenspiele und andere didaktische Elemente in der netzgestützten Hochschullehre. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 90–100. Ibidem, Stuttgart.

BUB (2007). Auszeichnung. Hamburg holt mit Chatbot Stella den Innovationspreis. *BUB – Forum Bibliothek und Information*, 59:20.

Budin, G. and Wright, S. E. (1997). *Handbook of Terminology Management*, volume 1 (Basic Aspects of Terminology Management). John Benjamins, Amsterdam.

Cadiz, J., Burkhalter, B., and Smith, M. (2000). Conversation trees and threaded chats. Online unter http://research.microsoft.com/research/pubs/view.aspx?tr_id=380.

Cassel, J. (2003). Mehr als nur ein nettes Gesicht: Embodied conversational interface agents. In Lindner, C., editor, *Avatare. Digitale Sprecher für Business und Marketing*, pages 249–265. Springer, Berlin/ Heidelberg/ New York.

Chen, D. and Sun, C. (2007). Realtime text chat via collaborative editing systems. Online unter http://www.cit.gu.edu.au/ david/CSCW02CEW/submissions/Chen.pdf.

Crystal, D. (2006). *Language and the Internet*. Cambridge University Press, Cambridge, 2nd edition.

Cyranek, G. (1988). Menschliche kommunikation und rechnerdialog. In Nullmeiner, E. and Rödiger, K.-H., editors, *Dialogsysteme in der Arbeitswelt*, pages 139–154. BI-Wissenschaftsverlag, Mannheim/ Wien/ Zürich.

Darhower, M. (2002). Interactional features of synchronous computer-mediated communication in the intermediate L2 class: A sociocultural case study. *CALICO Journal*, 19(2):249–277.

Davidson, P. (2005). This smartbot is a real smartass. *Wired*, 13(2).

Diekmannshenke, H. (2005). Politische Kommunikation in Zeiten des Internet. Kommunikationswandel am Beispiel moderierter und unmoderierter Politik-Chats. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 119–144. Ibidem, Stuttgart.

Döring, N. and Pöschl, S. (2005). Wissenskommunikation in themenbezogenen Online-Chats. Eine empirische Analyse von drei IRC-Channels zu Computerthemen. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 145–160. Ibidem, Stuttgart.

Drossmann, C. (2001). GermanAlice-D. Online unter http://www.alicebot.org/downloads/aiml/GermanA
standalone.zip.

Drucker, S. M., Farnham, S. D., and Smith, M. A. (2000). The social life of small graphical chat spaces. In *Proceedings of CHI 2000*.

Filinski, P. (1998). *Chatten in der CyberWorld*. VMI Buch AG, Bonn.

Forstliche Versuchs- und Forschungsanstalt Baden-Württemberg (2007). Zukunftsorientierte Waldwirtschaft. Online unter http://www.zukunftswald.de/page.php.

Geeb, F. (1998). *Semantische und enzyklopädische Informationen in Fachwörterbüchern. Eine Untersuchung zu fachinformativen Informationstypen mit besonderer Berücksichtigung wortgebundener Darstellungsformen*. Wirtschaftsuniversitat Aarhus, Århus.

Geeb, F. (2003). Lexikographische Informationsstrukturierung mit XML. *Information Wissenschaft & Praxis*, 7:415–421.

Geeb, F. (2006). lookedup4you – studerende, kunder og online-leksikografi. In Hansen, C., Lorentzen, H., and Trap-Jensen, L., editors, *Nordiske Studier i Leksikografi 8. Rapport fra Konference om Leksikografi i Norden*.

Geeb, F. and Spree, U. (2005). lookedup4you – mikrostruktur und makrostruktur und all das. produktion eines online-nachschlagewerks als studentisches projekt. ein erfahrungsbericht. *Information – Wissenschaft & Praxis*, 3:133–142.

Harnoncourt, M., Holzhauser, A., Seethaler, U., and Meinl, P. (2005). Referenzierbarkeit als Schlüssel zum effizienten Chat. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 160–179. Ibidem, Stuttgart.

Hentschläger, U. and Wiener, Z. (2002). *Webdramaturgie. Das audio-visuelle Gesamterlebnis – 3D, Streaming, Flash*. Markt+Technik, München.

Holmer, T. and Wessner, M. (2005). Gestaltung von Chat-Werkzeugen zur Verringerung der Inkohärenz. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 181–199. Ibidem, Stuttgart.

Holzhauser, A. (2003). Chat-Technologien in der Arbeitswelt. Online unter http://chat.themenplattform.com/126558.0/.

Husmann, H. (1998). *Chatten im Internet Relay Chat (IRC). Einführung und erste Analyse*. KoPäd, München.

Jia, J. (2007). The study of the application of a keywords-based chatbot system on the teaching of foreign languages. Online unter http://arxiv.org/abs/cs.CY/0310018/.

Kollock, P. and Smith, M. (1998). What do people do in virtual worls, an analysis of v-chat log file data. Online unter http://research.microsoft.com/scg/papers%5Ckollockv-chat.pdf.

Licklider, J. and Taylor, R. (1968). The computer as a communication device. *Science and Technology*, pages 21–31.

Lindner, C. (2003). Wer braucht wofür Avatare? In Lindner, C., editor, *Avatare. Digitale Sprecher für Business und Marketing*, pages 5–24. Springer, Berlin/ Heidelberg/ New York.

Medienpädagogischer Forschungsverbund Südwest (2007a). JIM-Studie 2006. Jugend, Information, (Multi-) Media. Online unter http://www.mpfs.de/fileadmin/JIM-pdf06/JIM-Studie_2006.pdf.

Medienpädagogischer Forschungsverbund Südwest (2007b). KIM-Studie 2006. Kinder + Medien, Computer + Internet. Online unter http://www.mpfs.de/fileadmin/KIM-pdf06/KIM2006.pdf.

Morphy, E. (2001). Jeeves exec explains self-service evolution. Online unter http://www.crm-daily.com/perl/story/18064.html.

MySQL AB (2007). MySQL 5.1 Reference Manual. 12.8 Full-Text Search Functions. Online unter http://dev.mysql.com/doc/refman/5.1/en/fulltext-search.html.

Nake, F. (1988). Dialogisieren mit dem computer - anmerkkungen zu entwicklung, begriff und technik der dialogsysteme. In Nullmeiner, E. and Rödiger, K.-H., editors, *Dialogsysteme in der Arbeitswelt*, pages 16–46. BI-Wissenschaftsverlag, Mannheim/ Wien/ Zürich.

Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann, San Francisco.

Nielsen, J. (2001). *Designing Web Usability*. Markt+Technik, München.

Niemann, F. (2003). Instant Messaging: Arbeitssitzung im Chat-Room. Online unter http://www.computerwoche.de/index.cfm?pid=816&pk=535484.

Puck, J. F. and Exter, A. (2005). Der Einsatz von Chats im Rahmen der Personalbeschaffung. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 289–301. Ibidem, Stuttgart.

Reeves, B. and Nass, C. (1996). *The media equation: how people treat computers, television and new media like real people and places*. University of Chicago Press.

Reichle, M. (2006). Entwicklung eines prototypischen Chatbots für die Universitätsbibliothek Hildesheim. In Womser-Hacker, C. and Mandl, T., editors, *Effektive Information Retrieval Verfahren in Theorie und Praxis: Proceedings Fünfter Hildesheimer Evaluierungs- und Retrieval (HIER) Workshop*, pages 43–46.

Runkehl, J., Schlobinsiki, P., and Siever, T. (1998). *Sprache und Kommunikation im Internet*. Vandenhoeck & Ruprecht, Opladen.

Salimi, M. (2003). Support-Chat und Avatare. In Lindner, C., editor, *Avatare. Digitale Sprecher für Business und Marketing*, pages 131–138. Springer, Berlin/ Heidelberg/ New York.

Samuelsen, P. (2003). Die Bedeutung von natürlichsprachlichen Dialogsystemen im Internet-Business. In Lindner, C., editor, *Avatare. Digitale Sprecher für Business und Marketing*, pages 27–35. Springer, Berlin/ Heidelberg/ New York.

Schmitt, A. A. (1983). *Dialogsysteme. Kommunikative Schnittstellen, Software-Ergonomie und Systemgestaltung*, volume 40 of *Informatik*. Bibliographisches Institut, Mannheim/ Wien/ Zürich.

Shneidermann, B. (1997). Direct manipulation versus agents: Paths to predictable, controllable and comprehensive interfaces. In Bradshaw, J., editor, *Software Agents*, pages 97–106. MIT Press.

Shuli, G. and Nielsen, J. (2002). Usability of websites for children: 70 design guidelines based on usability studies with kids. Usability report, Nielsen Norman Group, Freemont.

Staats- und Universitätsbibliothek (2005). Chatbot Stella. Online unter http://www.sub.uni-hamburg.de/informationen/projekte/infoass.html.

Stein, D. (2005). Distanz und Nähe in interaktiver Internetkommunikation. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 289–301. Ibidem, 339-347.

Storrer, A. (2001). Sprachliche Besonderheiten getippter Gespräche: Sprecherwechsel und sprachliches Zeigen in der Chat-Kommunikation. In Beißwenger, M., editor, *Chat-Kommunikation. Sprache, Interaktion, Sozialität und Identität in synchroner computervermittelter Kommunikation*, pages 3–24. Ibidem, Stuttgart.

Svensén, B. (2004). *Handbok i leksikografi*. Norstedts Akademiska Förlag, Stockholm.

Thommes, J. (2001). Die Chatsetter. *HORIZONT*, 19:78.

Trogemann, G. (2003). Mit Hand und Fuß. In Lindner, C., editor, *Avatare. Digitale Sprecher für Business und Marketing*, pages 269–290. Springer, Berlin/ Heidelberg/ New York.

Turney, P. D. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.

van Eckert, E. (2005). Termingebundene Chats one-to-one in der psychosozialen Beratung. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 349–359. Ibidem, Stuttgart.

Vetter, M. (2003). Qualitätskriterien für Online-Tests von Bots. In Lindner, C., editor, *Avatare. Digitale Sprecher für Business und Marketing*, pages 73–92. Springer, Berlin/ Heidelberg/ New York.

von Wendt, K.-L. (2003). Technische Grundlagen von natürlichsprachlichen Dialogsystemen. In Lindner, C., editor, *Avatare. Digitale Sprecher für Business und Marketing*, pages 39–47. Springer, Berlin/ Heidelberg/ New York.

Vronay, D., Smith, M., and Drucker, S. (2004). Alternative interfaces for chat. chat as a streaming media data type. Online unter http://research.microsoft.com/scg/papers/chat.htm.

Wallace, R. S. (2002). AIML overview. Online unter http://www.pandorabots.com/ pandora/pics/wallaceaimltutorial.html.

Weizenbaum, J. (1966). ELIZA – A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 10(8):36–45.

Wiegand, H. E. (1979). Definition und Terminologienormung – Kritik und Vorschläge. In Felber, H., Lang, F., and Wersig, G., editors, *Terminologie als angewandte Sprachwissenschaft. Gedenkschrift für E. Wüster*, pages 101–148. Saur, München/ New York/ London/ Paris.

Willand, I. (2002). *Chatroom statt Marktplatz. Identität und Kommunikation zwischen Öffentlichkeit und Privatheit*. KoPäd, München.

Wirth, T. (2003). Avatare und die Usability von Websites. In Lindner, C., editor, *Avatare. Digitale Sprecher für Business und Marketing*, pages 123–128. Springer, Berlin/ Heidelberg/ New York.

Wolff, P. (2003). Virtualisierung und Personalisierung – Technologietrends machen Avatare zur innovativen Mensch-Maschine-Schnittstelle. In Lindner, C., editor, *Avatare. Digitale Sprecher für Business und Marketing*, pages 293–299. Springer, Berlin/ Heidelberg/ New York.

Zumbach, J. and Spraul, P. (2005). Tutoring in synchronen Lernumgebungen. In Beißwenger, M. and Storrer, A., editors, *Chat-Kommunikation in Beruf, Bildung und Medien. Konzepte, Werkzeuge, Anwendungsfelder*, pages 379–391. Ibidem, Stuttgart.

Xiaofei Lu

A Hybrid Model for Chinese Word Segmentation

This paper describes a hybrid model that combines machine learning with linguistic and statistical heuristics for integrating unknown word identification with Chinese word segmentation. The model consists of two major components: a tagging component that annotates each character in a Chinese sentence with a position-of-character (POC) tag that indicates its position in a word, and a merging component that transforms a POC-tagged character sequence into a word-segmented sentence. The tagging component uses a support vector machine (Vapnik, 1995) based tagger to produce an initial tagging of the text and a transformation-based tagger (Brill, 1995) to improve the initial tagging. In addition to the POC tags assigned to the characters, the merging component incorporates a number of linguistic and statistical heuristics to detect words with regular internal structures, recognize long words, and filter non-words. Experiments show that, without resorting to a separate unknown word identification mechanism, the model achieves an F-score of 95.0% for word segmentation and a competitive recall of 74.8% for unknown word identification.

1 Introduction

Word segmentation is a prerequisite for all word-based Chinese language processing systems. For example, for speech synthesis, many phonological rules depend on correct word segmentation, and text-to-speech systems need to compute boundaries between intonational phrases in long utterances and assign relative prominence to words in those utterances (Sproat et al., 1996). For information retrieval, accurate word segmentation has been shown to measurably improve retrieval performance (Palmer and Burger, 1997; Foo and Li, 2004).

A Chinese word consists of one or more Chinese characters, or *zi*. In languages where word boundaries are clearly delimited by whitespace and punctuation marks, word segmentation is relatively straightforward. However, the absence of unambiguous word boundary markers in real Chinese text makes Chinese word segmentation a nontrivial challenge. The task is further complicated by the lack of a commonly accepted definition of word in Chinese among theoretical linguists. In practice, the Chinese language processing community adopts a pragmatic approach to this issue, where the definition of word varies depending on the purpose of the natural language processing system and the segmentation standard it adopts (Sproat et al., 1996; Wu, 2003; Gao et al., 2005).

A primary source of difficulty for Chinese word segmentation comes from segmentation ambiguities, including covering ambiguity and overlapping ambiguity (Liang, 1987). Covering ambiguity refers to the case where two segments may or may not be combined to form a larger segment. For example, the string 要害 may be segmented into two units 要/害 will/hurt 'will hurt' or one unit 要害 'vitals', depending on context. Overlapping ambiguity refers to the case where a segment may combine with either its preceding or following segment. For example, in the string 布什在谈话中指出, the segment 指 can potentially combine with either the preceding segment 中 or the following segment 出, as shown in (1) and (2) respectively. In this case, however, only the segmentation in (2) is acceptable.

* 布什	在	谈话	中指	出	(1)
Bush	at	talk	middle-finger	out	

布什	在	谈话	中	指出	
Bush	at	talk	middle	point-out	(2)
'Bush pointed out in his talk'					

Unknown words constitute a second source of difficulty for Chinese word segmentation. These are words that are not registered in the dictionary used by the word segmenter and/or are not found in the training data used to train the segmenter. While the size and domain specificity of the dictionary and training data may well affect the proportion of unknown words in real texts, unknown words will always exist, both because any dictionary creation effort has limited resources and because new words are constantly created. Chen and Bai (1998) report that 3.11% of the words in the Sinica Corpus (Chen et al., 1996), one of the largest word-segmented and POS-tagged Chinese corpora, are not listed in the CKIP lexicon, a Chinese lexicon with over 80,000 entries used for processing the corpus. These include unknown words of the categories of noun, verb, and adjective only, but not numeric type compounds or non-Chinese words. Xue (2003) partitions the 250K-word Penn Chinese Treebank (Xue et al., 2002) into training and test sets at a rather skewed ratio of 9.5:0.5 and finds that 4% of the words in the test set are unknown. Meng and Ip (1999) partition a smaller 72K-word corpus from Tsinghua University (Bai et al., 1992) at a 9:1 ratio, and report that 13% of the words in the test set are unknown.

Most previous studies treat word segmentation and unknown word identification as two separate problems, using a mechanism to identify unknown words in a post-processing step after word segmentation is done. However, determining where word boundaries are necessarily involves understanding how characters relate to and interact with each other in context, and it is desirable to capture this dynamic interaction by integrating unknown word identification with word segmentation. Several recent studies have taken a unified approach to unknown word identification and word segmentation (e.g., Sproat et al., 1996; Xue, 2003; Gao et al., 2005)

We describe a hybrid model that combines machine learning with linguistic and statistical heuristics for integrating unknown word identification with Chinese word segmentation. We adopt the notion of character-based tagging (Xue, 2003) to directly model the combinatory power of Chinese characters, i.e., the tendency for characters to combine with adjacent characters to form words, either known or unknown, in different contexts. The model consists of two components. First, a tagging component tags each character in a Chinese sentence with a position-of-character (POC) tag that indicates its position in a word. This component uses a support vector machine based tagger (Vapnik, 1995) to produce an initial tagging of the text and a transformation-based tagger (Brill, 1995) to improve the initial tagging. Second, a merging component transforms a POC-tagged character sequence into a word-segmented sentence, using a number of linguistic and statistical heuristics to detect words with regular internal structures, recognize long words, and filter non-words. Without resorting to a sophisticated mechanism for unknown word identification or additional resources other than a word-segmented training corpus, the model achieves an F-score of 95.0% for word segmentation and a recall of 74.8% for unknown word identification.

The rest of the paper is organized as follows. Section 2 reviews previous approaches to Chinese word segmentation. Section 3 describes the two components of the proposed model. Section 4 reports the experiment results of the model. Section 5 concludes the paper and points to avenues for future research.

2 Previous Approaches

Previous approaches for word segmentation generally fall into the following four categories: dictionary-based approaches, statistical approaches, statistical and dictionary-based approaches, and machine learning approaches.

In dictionary-based approaches, only words listed in the dictionary are identified. Most studies in this tradition use some variation of the maximum matching algorithm (e.g., Liang, 1987; Nie et al., 1995; Feng, 2001). Simply put, a maximum matching algorithm starts at the beginning of a sentence and inserts a word delimiter after the longest character string that matches a dictionary entry; it then moves to the character after the word delimiter and repeats the process until it reaches the end of the sentence. The algorithm can also start from the end of a sentence and search backwards. As the algorithm always favors the longest word, it in effect ignores segmentation ambiguities. To address this problem, some methods produce all possible segmentations first and then use certain criteria to select the best segmentation, e.g., using syntactic and semantic constraints (Yeh and Lee, 1991), favoring the segmentation in which each word has about the same length (Chen and Liu, 1992), etc. The performance of dictionary-based approaches is heavily dependent on the size and domain specificity of the dictionary used. These approaches also require the application of a separate unknown word identification mechanism in a post-processing step.

Relatively few studies have adopted purely statistical approaches to Chinese word segmentation. These studies use information-theoretical or probabilistic measures to determine whether adjacent characters form words or which segmentation is most likely for a sentence (e.g., Sproat and Shih, 1990; Ge et al., 1999). One advantage of statistical approaches is that they do not require any dictionary or word-segmented training data, but only a large raw corpus, which is relatively easy to obtain. However, they incorporate little linguistic knowledge and generally perform worse than other approaches. As Sproat et al. (1996) point out, such statistical methods can work as good unknown word identification mechanisms and can be used to augment existing electronic dictionaries and boost the performance of other word segmenters.

Statistical and dictionary-based approaches attempt to benefit from both worlds, using both information about words in the dictionary and statistical information derived from corpora to compute the most likely segmentation of a sentence. A good example of these approaches is Sproat et al. (1996). They represent a dictionary as a weighted finite state transducer, where weights or costs for word-strings are estimated based on their frequency in a large corpus. For each input sentence, they choose the path with the lowest cost as the best segmentation. This model requires separate mechanisms to estimate the probabilities of different types of unknown words. The authors recognize that the quality of the base lexicon is more important than the model and that unknown words constitute the greatest challenge.

With the availability of word-segmented training corpora, a number of supervised machine learning algorithms have been applied to Chinese word segmentation, including, for example, transformation-based learning (e.g., Hockenmaier and Brew, 1998; Florian and Ngai, 2001), maximum entropy (e.g., Xue, 2003; Low et al., 2005), conditional random fields (e.g., Tseng et al., 2005; Zhou et al., 2005), and linear mixture models (Gao et al., 2005). These approaches integrate unknown word identification with word segmentation and have achieved fairly competitive results.

3 Proposed Approach

This section describes a hybrid model that integrates unknown word identification with Chinese word segmentation. The major hypothesis tested in this study is that it is possible to directly model the combinatory power of individual characters, i.e., the tendency for individual characters to combine with adjacent characters to form words, either known or unknown, in different contexts. Furthermore, such modeling should have good potential for integrating unknown word identification with Chinese word segmentation. To this end, we adopt the notion of character-based tagging that has been employed for Chinese word segmentation and/or unknown word identification in a few recent studies (Zhang et al., 2002; Goh et al., 2003; Xue, 2003).

The proposed model consists of two major components. First, a tagging component annotates each character in a character sequence with a position-of-character (POC) tag that indicates the position of the character in a word. This component is based on the transformation-based learning (TBL) algorithm , using a tagger based on support vector machines (SVMs) to produce an initial tagging of a character sequence. Second, a merging component transforms the output of the tagging component, i.e., a POC-tagged character sequence, into a word-segmented sentence. In addition to the POC tags assigned to the character sequence, the merging component also makes use of a number of linguistic and statistical heuristics generalized from the training data to detect words with regular internal structures, recognize long words, and filter non-words. The overall architecture of the system is represented in Figure 1. An input Chinese sentence is first segmented into a character sequence, with a boundary marker after each character. The segmented character sequence is then processed by the POC-tagging component, where it is tagged first by the SVM-based initial tagger and then by the TBL tagger. Finally, the POC-tagged character sequence is transformed into a word-segmented sentence by the merging component.

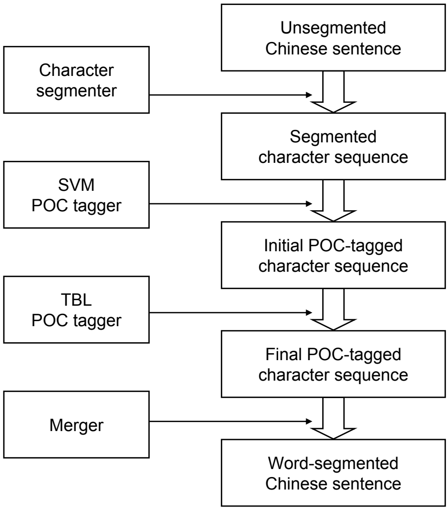


Figure 1: System architecture.

3.1 The POC Tagging Component

Table 1 summarizes the tagset defined for the POC tagging component. The tagset consists of four tags: L, M, R, and W, each of which indicates that the character is in a word-initial, word-middle, or word-final position or is a monosyllabic word.

A TBL Tagger. The transformation-based learning algorithm is adopted for the POC tagging component. This is done because, as Brill (1995) argues, it captures linguistic knowledge in a more explicit and direct fashion without compromising performance, an advantage over other promising, statistical machine learning algorithms. The implementation of the algorithm used in this task is fnTBL (Ngai and Florian, 2001), which is more efficient than Brill's original implementation.

The TBL algorithm requires a POC-tagged training corpus (the *truth*) and its corresponding raw version. A POC-tagged corpus can be converted from a word-segmented

Tag	Description
L	Character in word-initial position
M	Character in word-middle position
R	Character in word-final position
W	Monosyllabic word

Table 1: POC tagset.

corpus by assigning each character a POC tag based on its position in the word containing it. The conversion process is illustrated by the following example, where the word-segmented sentence in (3) is converted into a POC-tagged character sequence in (4).

今天 是 星期一 .
Today is Monday . (3)
'Today is Monday.'

今/L 天/R 是/W 星/L 期/M 一/R ./W (4)

In addition to the POC-tagged training corpus and its corresponding raw corpus, the algorithm requires three crucial components. The first is some initial tagging of the raw training corpus produced by an initial tagger. The algorithm itself places no requirement on the initial tagger. For example, it could assign the tag W (monosyllabic word) to each character in the corpus; or it could tag each character with its most likely tag derived from the training corpus. However, there is nothing that prevents one from using a more sophisticated tagger, and previous studies have shown that a better initial tagger leads to better final results and shorter learning time for the algorithm (Hockenmaier and Brew, 1998). For this reason, this study uses a more sophisticated initial tagger, an SVM-based tagger. A second initial tagger based on the hidden Markov model (HMM) is used as a comparison. Both of the two initial taggers are discussed in detail below.

The second component required by the TBL algorithm is the space of transformations allowed. The training process produces an ordered list of rules that can be applied to new text for tagging, but the kinds of transformations that can be learned need to be specified. Each transformation consists of a rewrite rule and a triggering environment. The set of transformations used in this study is similar to the set Ngai and Florian (2001) define for the task of base noun phrase chunking. In this case, however, the triggering context is defined over characters and POC tags instead of words and part-of-speech (POS) tags. The triggering context considered include the character and tag in the current position and the characters and tags in the three positions immediately preceding or following the current position. The specific rule templates defined are given in (5), where t_0 and c_0 denote the current tag and character respectively; t_{-i} and t_i denote the ith tag preceding or following t_0; c_{-i} and c_i denote the ith character preceding or following c_0; x, y, and z are tag variables; l, m, and n are character variables. The templates are categorized into three groups, depending on whether they require the identity of the current tag or character.

Change t_0 to Tag_j if (5)

1. $t_0 = x$, $c_0 = l$, and one of the following is true:
   a) $c_{-1} = m$
   b) $c_1 = m$
   c) $c_{-2} = n$
   d) $c_2 = n$
   e) $c_{-1} = m$ and $c_{-2} = n$
   f) $c_1 = m$ and $c_2 = n$
   g) $c_{-1} = m$ and $c_1 = n$
   h) $t_{-1} = y$
   i) $t_1 = y$
   j) $t_{-2} = z$
   k) $t_2 = z$
   l) $c_{-1} = m$ and $t_{-1} = y$
   m) $c_1 = m$ and $t_1 = y$
   n) $c_{-1} = m$ and $t_1 = y$
   o) $c_1 = m$ and $t_{-1} = y$
2. $t_0 = x$ and one of the following is true:
   a) $c_0 = l$
   b) $c_{-1} = m$
   c) $c_1 = m$
   d) $c_{-2} = n$
   e) $c_2 = n$
   f) $c_{-1} = m$ or $c_{-2} = m$
   g) $c_1 = m$ or $c_2 = m$
   h) $c_{-1} = m$ or $c_{-2} = m$ or $c_{-3} = m$
   i) $c_1 = m$ or $c_2 = m$ or $c_3 = m$
   j) $t_{-1} = y$
   k) $t_1 = y$
   l) $t_{-2} = z$
   m) $t_2 = z$
   n) $t_{-1} = y$ or $t_{-2} = y$
   o) $t_1 = y$ or $c_2 = y$
   p) $t_{-1} = y$ or $t_{-2} = y$ or $t_{-3} = y$
   q) $t_1 = y$ or $t_2 = y$ or $t_3 = y$
   r) $t_{-1} = y$ and $t_1 = z$
   s) $t-_1 = y$ and $t_{-2} = z$
   t) $t_1 = y$ and $t_2 = z$
   u) $c_1 = m$ and $t_1 = y$ and $t_2 = z$

3. $c_0 = l$ and one of the following is true:

   a) $t{-1} = y$ and $t_{-2} = z$
   b) $c{-1} = m$ and $t{-1} = y$
   c) $c_1 = m$ and $t_1 = y$

The third component of the algorithm is a scoring function, which is used to compare the corpus to the *truth* and determine which transformation should be learned. In this study, the scoring function used is the number of POC tagging error reductions achieved after applying a transformation.

Once all the components are in place, the training process is iterative and takes place as follows. At each iteration, the learner applies each possible instantiation of the transformation templates to the text (starting with the text tagged by the initial tagger), counts the number of tagging error reductions each transformation achieves, and chooses the transformation that achieves the greatest number of tagging error reductions. That transformation is then applied to the text, and the learning process repeats, until no more transformations reduce errors beyond a pre-determined threshold. The output of the algorithm is a ranked list of transformations. Once the system is trained, a new sentence is tagged by first applying the initial tagger and then by applying the learned transformations to the sentence in the right order.

An SVM-Based Initial Tagger. SVMs (Vapnik, 1995) are binary classifiers on a feature vector space $\Re^L$. Given a set of training data, $\{(x_i\ ,y_i)|x_i{\in}\Re^L,\ y_i{\in}\{\pm1\},\ 1{\leq}i{\leq}l\}$, where x_i is the ith sample in the training data and y_i is the label of x_i, a hyperplane, given in (6), separates the set into two classes in such a way that the constraints in (7) are satisfied:

$$w \cdot x + b = 0, w \in \Re^L, b \in \Re \tag{6}$$

$$y_i \cdot (w \cdot x_i + b) \geq 1 \tag{7}$$

In (6) and (7), w is a weight vector with one weight for each feature, and b is a bias, which is the distance of the hyperplane to the origin. Among all hyperplanes that separate the training data into two sets, SVMs find the optimal hyperplane with maximal margin, i.e., maximal distance between the hyperplane and the nearest positive and negative samples, because it is expected to minimize expected test error. Giving a test example x, its label y is determined by the sign of a discrimination function $f(x)$ given by the SVMs classifier as follows:

$$f(x) = sgn(\sum_{z_i \in SV} \alpha_i y_i K(x, z_i) + b) \tag{8}$$

where $b \in \Re$, z_i is a support vector, which receives a non-zero weight α_i, $K(x,\ z_i)$ is a polynomial kernel function of degree d given by $K(x, z_i)$ = $(x \cdot z_i$ + $1)^d$, which maps vectors into a higher dimensional space where all combinations of up to d features are considered, and SV denotes the set of support vectors, i.e., the vectors that receive a non-zero weight. The support vectors and the parameters are determined by quadratic

Characters	$c_{-2}, c_{-1}, c_0, c_1, c_2$
POC tags	t_{-2}, t_{-1}
Ambiguity class	a_0, a_1, a_2
Character bigrams	$(c_{-2}, c_{-1}), (c_{-1}, c_1), (c_{-1}, c_0), (c_0, c_1), (c_1, c_2)$
POC bigrams	$(t_{-2}, t_{-1}), (t_{-1}, a_1), (a_1, a_2)$
Character trigrams	$(c_{-2}, c_{-1}, c_0), (c_{-2}, c_{-1}, c_1), (c_{-1}, c_0, c_1), (c_0, c_1, c_2), (c_1, c_1, c_2)$
POC trigrams	$(t_{-2}, t_{-1}, a_0), (t_{-2}, t_{-1}, a_1), (t_1, a_0, a_1), (t_{-1}, a_1, a_2)$

Table 2: Feature set for the SVM-based POC tagger.

programming. If $f(x) = +1$, then x is a positive member, otherwise it is a negative member.

The implementation of the SVM classifier used in this study is SVMTool (Giménez and Màrquez, 2004). As SVMs are binary classifiers, some adaptation is necessary to make them suitable for multi-class classification tasks. Giménez and Màrquez use a one-per-class binarization, where they train an SVM for each class to determine whether an example is of this class or not. At classification time, the most confident tag given by all SVMs is selected.

The features used for POC tagging include character and POC tag n-grams. The two characters and POC tags preceding and following the current character and POC tag are considered. At running time, the POC tags of the characters to the right of the current character are not known. In SVMTool, a general ambiguity class tag, i.e., a label that concatenates all the possible POC tags for a character, is used for the right context characters. Table 2 summarizes the feature set used for POC tagging. For unknown characters, i.e., characters that are not found in the training data, only the POC features are used.

An HMM-Based Initial Tagger. To compare the impact of the initial tagger on the performance of the TBL algorithm, we implement a simple first-order HMM tagger (Charniak et al., 1993). This model computes the most likely tag sequence $t_1...t_n$ for a character sequence $c_1...c_n$, as follows:

$$\arg\max_{t_1...t_n} \prod_{i=1}^{n} p(t_i|t_{i-1}) p(t_i|c_i) \qquad (9)$$

where t_i and c_i denote the ith tag in the tag sequence and the ith character in the character sequence respectively, $p(t_i|t_{i-1})$ denotes the transition probability, i.e., the probability of a tag given its previous tag, and $p(t_i|c_i)$ denotes the lexical probability, i.e., the probability of a tag given a character. The transition and lexical probabilities are estimated from the training corpus. For unknown characters, the lexical probabilities are uniformly distributed among the four POC tags defined in the tagset.

The transition probabilities are not smoothed, as all unseen POC tag bigrams are impossible combinations. Once the model is trained, the Viterbi algorithm (Rabiner, 1989) is used to tag new text.

3.2 The Merging Component

The output of the tagging component, i.e., a POC-tagged character sequence, becomes the input of the merging component. The merging component transforms the POC-tagged character sequence into a word-segmented sentence. By default, it concatenates all the characters in the character sequence, inserting a word boundary marker after each character tagged *R* (word-final position) or *W* (monosyllabic word). In addition to the POC tags assigned to the character sequence, the merging component also makes use of a number of linguistic and statistical heuristics for 1) detecting various types of unknown words with regular internal structures, i.e., numeric type compounds and non-Chinese words, reduplicated and derived words, and transliterated foreign names; 2) recognizing long words that have appeared in the training corpus; and 3) filtering non-words. The specific heuristics used in the merging component are detailed below.

Numeric Type Compounds and Non-Chinese Strings. The merging component uses regular expressions to detect numeric type compounds and non-Chinese character strings. The types of numeric type compounds handled using regular expressions include dates, times, percentages, fractions, numbers, etc. Non-Chinese character strings include email addresses, words or acronyms in foreign alphabets, etc. The patterns for each of these types of words are generalized from the training corpus. A character string that fits one of these patterns is grouped into one segmentation unit, and a word boundary marker is inserted after it. Gao et al. (2005) show that the performance of detecting such words using their internal properties alone is comparable with that of using contextual information.

Reduplicated and Derived Words. The merging component also uses heuristics to detect reduplicated words that are three or more characters long and a few types of derived words with predictable internal structures. In Chinese, many monosyllabic and disyllabic words can reduplicate in various patterns, e.g., 读书 'read' and 读读书 'read a little' (AAB pattern), and 高兴 'happy' and 高高兴兴 'very happy' (AABB pattern). If a string of characters fits one of the reduplication patterns in Chinese, it is grouped as one word.

For derived words, many morphemes are ambiguous between an affix and a word. For example, the morpheme 家 can be either an affix '-ist' or a common noun 'family'. Detection of the correct use of such ambiguous morphemes can be especially hard in disyllabic words. To avoid overgeneration, the heuristics only attempt to detect derived words formed with a root word that is two or more characters long and an unambiguous affix, such as 学家 study-expert '-ist'. If an unambiguous prefix or suffix is detected, it is attached to its following or previous word, if that word is at least two characters long.

Transliterated Foreign Names. Foreign names are usually transliterated using a sequence of Chinese characters, and most transliterated foreign names are three or more characters long. The error analysis of the results of the default merger indicates that these words pose a challenge to the POC tagger. Since only a subset of Chinese characters are used in transliterations, it is possible to use heuristics to detect a substantial part of transliterated names.

A list of characters used for transliterations is acquired from the wordlist generated from the training corpus. The algorithm starts with a simple observation, i.e., there is a dot used exclusively within transliterated foreign names to indicate different parts of a name, such as first, middle, or last name. For example, 'George Bush' is transliterated as 乔治·布什, where the dot within the name indicates that 乔治 'George' and 布什 'Bush' are different parts of the same name. Based on this observation, the algorithm acquires a list of characters for transliterations in the following three steps:

1) Extract names with the dot from the wordlist and store all characters in these names in a seed list.

2) Extract all candidate names that are four or more characters long with all but one character from the seed list. For each candidate, add the one character not on the seed list to the seed list. Repeat until no more characters can be acquired.

3) Filter out characters acquired in step 2 that have appeared only in one candidate.

In processing new text, the algorithm first uses the final character list to identify candidate names and then filters candidates using contextual information, as follows:

1) Add characters immediately before or after the dot in the text to the final list of characters, if they are not already included.

2) Identify character n-grams ($n \geq 3$) whose component characters are all from the list. For each n-gram, do steps 3-7.

3) Extract 5 characters to its left and right.

4) Strip the leftmost character of the n-gram if it forms a word with the 1, 2, 3, 4, or 5 left context characters and add it to the left context. Iterate until the leftmost character of the n-gram no longer forms a word with its left context characters.

5) Strip the leftmost 2 characters of the n-gram if they form a word with the 1, 2, 3, 4, or 5 left context characters and add them to the left context. Iterate until the leftmost 2 characters of the n-gram no longer form a word with its left context characters.

6) Do steps 4 and 5 with the rightmost characters of the n-gram and the right context characters.

7) If the final n-gram has three or more characters, it is considered a transliterated foreign name.

Long Words. In addition to the three types of heuristics used to detect unknown words with regular internal structures discussed above, the merging component also uses heuristics to detect known long words. One hypothesis tested in this research is that longer words behave more consistently and introduce less ambiguity than shorter words. In particular, we hypothesize that if a string of four or more characters has appeared in the training corpus as a word, it is also likely to be a word in the test corpus. Whereas there are foreseeable counterexamples for this hypothesis, its usefulness will be empirically tested in the experiments. There is no dictionary that is adopted across all segmentation standards and corpus annotation projects. Given the fuzzy line between compounds and phrases in Chinese, a character string may be considered as a compound in one segmentation standard but a phrase consisting of two or more words in another segmentation standard. For example, the string 个人所得税 'personal income tax' may be considered one word or three words, 个人 'personal', 所得 'income', and 税 'tax', depending on the definition of the word in the segmentation standard. Within the same segmentation standard, however, such strings should be treated consistently. To adapt to the segmentation standard following which the corpus is segmented, the merging component uses the wordlist generated from the segmented training corpus instead of any existing dictionary. If a string of four or more characters is found in the wordlist, the merging component considers it as one word.

Non-Word Filtering. The last type of heuristics is for filtering non-words, i.e., false new word candidates. The first of these is used to detect bisyllabic non-words. If the POC tagger tags two adjacent characters as L (word-initial) and R (word-final), the string is a candidate new word, if it has not occurred in the training data. The candidate is filtered in two steps.

First, it is checked against two short lists of characters that contain frequent, unproductive morphemes that do not occur in 1) word-initial position of new bisyllabic words, e.g., 而 'but', or 2) word-final position of new bisyllabic words, e.g., 这 'this', respectively. If the first or second character of the candidate is on the first or second list, respectively, the candidate is split into two words.

Second, the probability for a character to appear in word-initial or word-final position is estimated from the training data as in (10):

$$P(C, Pos) = \frac{F(C, Pos)}{F(C)} \tag{10}$$

where C denotes a character, Pos denotes a position in a word, $P(C,Pos)$ denotes the probability that C occurs in Pos, $F(C,Pos)$ denotes the number of times C occurs in Pos, and $F(C)$ denotes the number of times C appears in any position of any word. Given a candidate new word, the probability for it to be a word is computed as the joint probability of $P(C,Pos)$ for both of its component characters. If the joint probability is below a pre-determined threshold, then the candidate is considered a non-word and is split into two words.

Finally, the following two rules are used to merge or split character strings of certain patterns, based on the error analysis in the development stage. First, if two adjacent characters are both tagged *W* (monosyllabic word) but have always occurred in the training data as a word, they are merged into a word. Second, if three adjacent characters are tagged *W* (monosyllabic word), *L* (word-initial), and *R* (word-final) respectively, but the first two characters form a known word and the last two characters do not, then the first two are grouped into a word and the last one is left as a monosyllabic word.

4 Results

The model is developed and tested using the Contemporary Chinese Corpus developed at Peking University in mainland China (Yu et al., 2002). This corpus contains all the news articles published in January, 1999 in *People's Daily*, a major newspaper in mainland China. The corpus has a total of over 1.12 million tokens and is word-segmented and POS-tagged. The corpus is randomly partitioned into three sets, with 80% used for training, 10% used for development, and 10% reserved for testing. The final model is trained on the union of the training and development sets and results are reported on the test set.

4.1 POC Tagging Results

As discussed earlier, two initial taggers are used to compare the impact of the initial tagger on the TBL algorithm, namely, a first-order HMM tagger and an SVM-based tagger. In both cases, the same rule templates described in section 3.1 are used. The threshold for the scoring function of the TBL algorithm is set to 1, i.e., all rules that achieve two or more tagging error reductions are learned. In addition, fnTBL makes it possible to learn rules that, at the end of the training process, result in no negative application but a number of positive applications greater than a pre-determined threshold. This threshold is set to 1 as well, i.e., all rules that achieve two or more positive applications with no negative application at the end of the training process are also learned. Both thresholds are set empirically in the development stage. The results of the two initial taggers as well as the improved results achieved by the TBL algorithm over their output are summarized in Table 3.

POC Tagger	Accuracy
HMM tagger	0.814
HMM + TBL	0.936
SVM tagger	0.931
SVM + TBL	0.946

Table 3: POC tagging results.

The HMM tagger achieves an accuracy of 81.4%, which is improved to 93.6% by the TBL algorithm. This amounts to an absolute accuracy improvement of 12.2%, which equals to an impressive tagging error reduction rate of 65.6%. The SVM-based tagger achieves a much better initial tagging than the HMM tagger, with an accuracy of 93.1%. The TBL algorithm achieves an absolute accuracy improvement of 1.5%, or a tagging error reduction rate of 21.7%. The better initial tagging achieved by the SVM-based tagger results in less improvement for the TBL algorithm, but better overall tagging accuracy.

4.2 Segmentation Results

The output of the merging component is evaluated using the scoring algorithm adopted for the SIGHAN Chinese segmentation bakeoffs (Sproat and Emerson, 2003; Emerson, 2005). This algorithm evaluates word segmenters in terms of recall, precision, F-score, recall for out-of-vocabulary (OOV) words (i.e., unknown words), and recall for in-vocabulary (IV) words.

The results for word segmentation and unknown word identification are summarized in Table 4. As shown in the second row of the figure, the default merger, which uses the POC tags assigned to the character string alone without resorting to any heuristics, achieves an F-score of 93.5% along with a recall rate of 73.8% for unknown word identification. Rows three through five indicate the improvement achieved with each of the three types of heuristics incorporated in the merging component. Row three shows that the heuristics for identifying unknown words with regular internal structures (UWI) improve the recall rate for unknown words (R_{OOV}) by 2.2%; row four shows that the heuristics for long word identification (LWI) slightly boost performance on known words; and row five shows that the heuristics for non-word filtering (NWF) improve performance on known words, although the recall rate for unknown word identification is brought down by 1.2%. When all the heuristics are used together with the POC tags, the F-score for word segmentation is improved by 1.5% to 95.0%, and the recall rate for unknown word identification is also improved by 1% to 74.8%.

Merger	R	P	F	R_{OOV}	R_{IV}
Default	0.932	0.938	0.935	0.738	0.944
+UWI	0.933	0.939	0.936	0.760	0.944
+LWI	0.933	0.940	0.937	0.737	0.945
+NWF	0.942	0.944	0.943	0.726	0.955
+ALL	0.947	0.952	0.950	0.748	0.959

Table 4: Word segmentation results.

Since the model uses no additional resources other than the training data, the results are directly comparable with the results of the systems that participated in the closed track of the Peking University Corpus in the second SIGHAN Chinese Segmentation

Bakeoff. Table 5 summarizes the results of our model and the top three systems in the bakeoff, namely, Chen et al. (2005), Tseng et al. (2005), and Zhang et al. (2005). As these results indicate, without a complicated unknown word recognition mechanism, the final model performs at the state of the art for word segmentation along with a competitive recall rate for unknown word identification.

System	R	P	F	R_{OOV}	R_{IV}
Ours	0.947	0.952	0.950	0.748	0.959
Chen	0.953	0.946	0.950	0.636	0.972
Tseng	0.946	0.954	0.950	0.787	0.956
Zhang	0.952	0.945	0.949	0.673	0.969

Table 5: Comparison of word segmentation results.

5 Discussion and Conclusion

This paper presents a new hybrid model that integrates unknown word identification with Chinese word segmentation using the notion of character-based tagging. The major hypothesis tested in this model is that character-based tagging has good potential for integrating the two, as it allows one to directly model the tendency for individual characters to combine with adjacent characters to form words in different contexts, regardless of whether the words formed are known or unknown. This hypothesis is confirmed by the results. With the default merger, the model achieves a competitive rate for unknown word identification without resorting to any complicated unknown word recognition mechanism.

One advantage of the use of the notion of character-based tagging is that since it is essentially a classification problem, it can directly benefit from improvements in classifiers. A second advantage of the two-component setup of the model is that it allows easy integration of additional linguistic insights as well as statistical heuristics in the merging stage. The current merging component of the model incorporates three types of linguistic and statistical heuristics. The heuristics for the recognition of unknown words with regular or predictable internal properties, including numeric type compounds, reduplicated words and certain derived words, and transliterated foreign names, prove useful for enhancing the performance on unknown words. The heuristics for the recognition of known long words tests the hypothesis that long words behave more consistently and introduce less segmentation ambiguity than shorter words. The hypothesis is supported by the improvement achieved by the heuristics. Finally, the heuristics for non-word filtering successfully catch a substantial portion of false unknown words detected by the POC tagger.

The current model does not make use of any additional resources other than the training data. Various lexical resources have been used in different word segmentation systems. For example, some of the resources used in Gao et al. (2005), which is by far the most sophisticated word segmentation system with access to the richest resources, include a 98,668-entry lexicon, a semi-automatically compiled morpho-lexicon that contains 59,960 morphologically derived words with information about morphological patterns and stems for each entry, a list of 373 family name characters, a list of 30,000 location names, a list of 1,355 organization names, a list of 618 transliterated name characters, a list of 151 characters for single-character person names, and a list of 177 single-character location names. Such lexical resources can be used to improve the tagging component by enriching the feature set used for POC tagging. They can also be used to improve and enrich the heuristics used in the merging component.

References

Bai, S. H., Xia, Y., and Huang, C. (1992). Automatic part of speech tagging system for Chinese. Technical report, Tsinghua University, Beijing, China.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

Chen, A., Zhou, Y., Zhang, A., and Sun, G. (2005). Unigram language model for Chinese word segmentation. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*, pages 138–141.

Chen, K.-J. and Bai, M.-H. (1998). Unknown word detection for Chinese by a corpus-based learning method. *International Journal of Computational Linguistics and Chinese Language Processing*, 3(1):27–44.

Chen, K.-J., Huang, C.-R., Chang, L.-P., and Hsu, H.-L. (1996). Sinica corpus: Design methodology for balanced corpora. In *Proceedings of the 11th Pacific Asia Conference on Language, Information, and Communication*, pages 167–176.

Chen, K.-J. and Liu, S.-H. (1992). Word identification for Mandarin Chinese sentences. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 101–107.

Emerson, T. (2005). The second Chinese word segmentation bakeoff. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*, pages 123–133.

Feng, Z. (2001). Hybrid approaches for automatic segmentation and annotation of a Chinese text corpus. *International Journal of Corpus Linguistics*, 6:35–42.

Florian, R. and Ngai, G. (2001). Multidimensional transformation-based learning. In *Proceedings of the 5th Workshop on Computational Language Learning*, pages 1–8.

Foo, S. and Li, H. (2004). Unsupervised Chinese word segmentation and its effect on information retrieval. *Information Processing and Management: An International Journal*, 40(1):161–190.

Gao, J., Li, M., Wu, A., and Huang, C.-N. (2005). Chinese word segmentation and named entity recognition: a pragmatic approach. *Computational Linguistics*, 31(4):531–574.

Ge, X., Pratt, W., and Smyth, P. (1999). Discovering Chinese words from unsegmented text. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 271–272.

Giménez, J. and Màrquez, L. (2004). SVMTool: A general POS tagger generator based on support vector machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 43–46.

Goh, C.-L., Asshara, M., and Matsumoto, Y. (2003). Chinese unknown word identification using character-based tagging and chunking. In *Proceedings of the Interactive Posters and Demonstration Session at the 41st Annual Meeting of the Association for Computational Linguistics*, pages 197–200.

Hockenmaier, J. and Brew, C. (1998). Error-driven segmentation of Chinese. In *Proceedings of the 12th Pacific Asia Conference on Language, Information and Computation*, pages 218–229.

Liang, N. (1987). Shumian hanyu zidong fenci xitong-CDWS [an automatic segmentation system for written Chinese-CDWS]. *Journal of Chinese Information Processing*, 1(2):44–52.

Low, J., Ng, H., and Guo, W. (2005). A maximum entropy approach to Chinese word segmentation. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*, pages 161–164.

Meng, H. and Ip, C. (1999). An analytical study of transformational tagging for Chinese text. In *Proceedings of the 12th Research on Computational Linguistics Conference*, pages 101–122.

Ngai, G. and Florian, R. (2001). Transformation-based learning in the fast lane. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 40–47.

Nie, J.-Y., Hannan, M.-L., and Jin, W. (1995). Unknown word detection and segmentation of Chinese using statistical and heuristic knowledge. *Communications of COLIPS*, 5(1/2):47–57.

Palmer, D. and Burger, J. (1997). Chinese word segmentation and information retrieval. In *Proceedings of the 1997 AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, pages 175–178.

Rabiner, L. (1989). A tutorial of hidden Markov models and selected applications in speech recognition. In *Proceedings of IEEE 1989*, pages 257–286.

Sproat, R. and Emerson, T. (2003). The first international Chinese word segmentation bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*.

Sproat, R. and Shih, C. (1990). A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, 4(4):336–351.

Sproat, R., Shih, C., Gale, W., and Chang, N. (1996). A stochastic finite-state word segmentation algorithm for Chinese. *Computational Linguistics*, 22(3):377–404.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., and Manning, C. (2005). A conditional random fields word segmenter for SIGHAN Bakeoff 2005. In *Proceedings of the 4th SIG-HAN Workshop on Chinese Language Processing*, pages 168–171.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin.

Wu, A. (2003). Customizable segmentation of morphologically derived words in Chinese. *International Journal of Computational Linguistics and Chinese Language Processing*, 8(1):1–28.

Xue, N. (2003). Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

Xue, N., Chiou, F.-D., and Palmer, M. (2002). Building a large annotated Chinese corpus. In *Proceedings of the 19th International Conference on Computational Linguistics*.

Yeh, C.-L. and Lee, H.-J. (1991). Rule-based word identification for Mandarin Chinese sentences - a unification approach. *Computer Processing of Chinese and Oriental Languages*, 5(2):97–118.

Yu, S., Duan, H., Zhu, X., and Sun, B. (2002). The basic processing of contemporary Chinese corpus at Peking University. *Journal of Chinese Information Processing*, 16(5):49–64.

Zhang, H., Liu, T., Ma, J., and Liao, X. (2005). Chinese word segmentation with multiple post-processors in HIT-IR Lab. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*, pages 172–175.

Zhang, K., Liu, Q., Zhang, H., and Cheng, X. (2002). Automatic recognition of Chinese unknown words based on roles tagging. In *Proceedings of the 1st SIGHAN Workshop on Chinese Language Processing*, pages 71–78.

Zhou, J., Ni, B., and Chen, J. (2005). A hybrid approach to Chinese word segmentation around CRFs. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*, pages 196–199.

Eric Atwell
School of Computing
University of Leeds
Leeds LS2 9JT
United Kingdom
eric@comp.leeds.ac.uk

Gerhard Paaß
Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS
Abteilung KD
Schloss Birlinghoven
53754 Sankt Augustin
gerhard.paass@iais.fraunhofer.de

Bayan Abu Shawar
Information Technology Department
Arab Open University / Jordan Branch
Amman 11953
Jordan
b_shawar@aou.edu.jo

Edda Leopold
Hochschule für Angewandte Wissenschaften
Hamburg
Stiftstraße 69
20099 Hamburg
leopold@mt.haw-hamburg.de

Franziskus Geeb
Hochschule für Angewandte Wissenschaften Hamburg
Fakultät Technik und Informatik (TI)
Berliner Tor 7
20099 Hamburg
franziskus.geeb@bui.haw-hamburg.de

Xiaofei Lu
Pennsylvania State University
301 Sparks Building
University Park, PA 16802
USA
xxl13@psu.edu

Jörg Kindermann
Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS
Abteilung KD
Schloss Birlinghoven
53754 Sankt Augustin
joerg.kindermann@iais.fraunhofer.de

Caroline Sporleder
Department of Language and Information
Science
Tilburg University
Warandelaan 2
5037 AB Tilburg
The Netherlands
csporled@uvt.nl