

The Workshop Programme

Agenda (Morning Session)

Technical Papers (8:00-9:20)

8:00-8:20 *'Multilingual Terminology Databanks for Web Mining'*
António Ribeiro, Universidade Nova de Lisboa

8:20-8:40 *'Grammar Learning by Partition Search'*
Anja Belz, ITRI University of Brighton

8:40-9:00 *'A Semantic-driven Approach to Hypertextual Authoring'*
R. Basili, A. Moschitti, M.T. Pazienza and F.M. Zanzotto, University of Rome, Tor Vergata

9:00-9:20 *'Advantages of ontology-based user profiling in the NAMIC project'*
Jan De Bo and Ben Majer, VUB STARLab, University of Brussels

9:20-10:05 Invited Talk *'Methods, representation and linguistic bias in Adaptive IE'*
Roberto Basili, University of Rome, Tor Vergata

10:05-10:25 Coffee

Technical Papers (10:25-11:45)

10:25-10:45 *'Description of Events: An Analysis of Keywords and Indexical Names'*
Khurshid Ahmad, Paulo C F de Oliveira, Pensiri Manomaisupat, Matthew Casey and Tugba Taskaya, University of Surrey

10:45-11:05 *'Learning IE patterns: a terminology extraction perspective.'*
Roberto Basili, Maria Teresa Pazienza, Fabio Massimo Zanzotto, University of Rome, Tor Vergata

11:05-11:25 *'Unsupervised Event Clustering in Multilingual News Streams'*
Martijn Spitters and Wessel Kraaij, Department of Multimedia Technology & Statistics, The Netherlands

11:25-11:45 *'Large-scale Multilingual Information Extraction'*
R. Catizone, A. Setzer and N. Webb, University of Sheffield

11:45-12:30 Invited Talk: *'User Driven Information Extraction for the Web'*
Fabio Cirevegna, University of Sheffield

12:30-1:30 Panel and Round Table on *Adaptive Technologies and their implications on advanced HLT applications (IR, IE, Q&A and KM)*

panelists:

Nino Varile (EC Commission)

F. Gardin (AISoftware)

Y. Wilks (Univeristy of Sheffield)

M.T. Pazienza (University of Rome, Tor Vergata)

Francesco Danza (Knowledge Stones S.p.A.)

Remi Zajac (Systran Software)

Workshop Organisers

Roberta Catizone, University of Sheffield
Roberto Basili, University of Rome, Tor Vergata
Maria-Teresa Pazienza, University of Rome, Tor Vergata
Maria-Vittoria Marabello, Knowledge Stones S.p.A., Rome

Workshop Programme Committee

Roberta Catizone	University of Sheffield
Walter Daelemans	CNTS/Language Technology Group, Antwerp
M. V. Marabello	KnowledgeStones S.p.A
M. T. Pazienza	University of Rome, Tor Vergata
G. Rigau	Polytechnical University of Catalonia
Horatio Rodriguez	Polytechnical University of Catalonia
A. Setzer	University of Sheffield
N. Webb	University of Sheffield
Y. Wilks	University of Sheffield
Rémi Zajac	Systran Software, CA
F.M. Zanzotto	University of Rome, Tor Vergata

Table of Contents

<i>Multilingual Terminology Databanks for Web Mining</i>	1
António Ribeiro	
<i>Grammar Learning by Partition Search</i>	9
Anja Belz	
<i>A Semantic-driven Approach to Hypertextual Authoring</i>	17
R. Basili, A. Moschitti, M.T. Pazienza and F.M. Zanzotto	
<i>Advantages of ontology-based user profiling in the NAMIC project</i>	23
Jan De Bo and Ben Majer	
<i>Description of Events: An Analysis of Keywords and Indexical Names</i>	29
A. Khurshid, P. C F de Oliveira, P. Manomaisupat, M. Casey and T. Taskaya	
<i>Learning IE patterns: a terminolgy extraction perspective</i>	36
Roberto Basili, Maria Teresa Pazienza and Fabio Massimo Zanzotto	
<i>Unsupervised Event Clustering in Multilingual News Streams</i>	42
Martijn Spitters and Wessel Kraaij	
<i>Large-scale Multilingual Information Extraction</i>	47
R. Catizone, A. Setzer and N. Webb	

Author Index

Ahmad, K.	29
Basili, R.	17, 36
Belz, A.	9
de Bo, J.	23
Casey, M.	29
Catizone, R.	47
Kraaij, W.	42
Majer, B.	23
Manomaisupat, P.	29
Moschitti, A.	17
de Oliveira, P. C F.....	29
Pazienza, M.T.	17, 36
Ribeiro, A.	1
Setzer, A.	47
Spitters, M.	42
Taskaya, T.	29
Webb, N.	47
Zanzotto, F.	17, 36

Multilingual Terminology Databanks for Web Mining

António Ribeiro*, Gabriel Lopes* and João Mexia⁺

Universidade Nova de Lisboa
Faculty of Sciences and Technology, Department of *Informatics/[†]Mathematics
Quinta da Torre, Monte da Caparica, P-2829-516 Caparica, Portugal
{ambar, gpl}@di.fct.unl.pt

Abstract

This paper presents on-going research on a methodology to build multilingual terminology databanks from parallel texts in several languages. Web mining is a potential application for these databanks for they allow not only multilingual document content modelling but also multilingual retrieval of documents matching a user's query in different languages. We start by describing a methodology to align parallel texts and to extract multiword term Translation Equivalents, using language independent and statistically supported techniques. Next, we present some approaches for multilingual mining in order to provide the context for this work. Finally, we discuss how these multilingual terminology databanks can be used in the framework of multilingual mining.

1. Introduction

In an increasingly multilingual Web, monolingual web search engines have become unable to mine the web and retrieve simultaneously documents written in different languages for a query made in a particular language. It may be the case that the most relevant documents are not written in the language the query was made; the user may not know what language to choose to retrieve the best documents. Current monolingual search engines cannot help on this. Thus, it would be wise to have multilingual web miners which would allow multilingual web searches and provide the user either the original document if the user understands the language it is written in or a translation of the document in a language selected by the user. Multilingual web search engines must be able to cope with Cross-Language Information Retrieval (CLIR) if they are to satisfy their customers. CLIR addresses precisely the possibility of making queries in one language and retrieving relevant documents in other languages (Brown *et al.*, 2000).

Google Inc., the company that owns the popular Google web search engine, has recently released a note stressing an increase in the number of web pages written in languages other than English: "Of the 2 billion web pages in Google's index [<http://www.google.com>], more than a quarter are in languages other than English" (Google Inc., 2001a). In fact, English is steadily becoming less *the* language of the Web. As more web pages are written in other languages, web searches are doomed to be confined to the documents written in the same language of the query if web search engines are not able to handle searches in multilingual documents. In a world which promotes information exchange, this seems to do the opposite through divisions and to raise the issue whether a 'Multilingual Information Society'¹ can actually be real.

Although users' experience says that they are better off with English for Web searches world wide, a press release also from Google Inc. (2001c) reports a growing trend in the number of web searches done in languages other than English in its own web search engine.

¹ This is the name of a programme supported by the European Commission, which aims at protecting and safeguarding pluralism, diversity and the principle of equality among all languages (<http://www.hltcentral.org/page-762.0.shtml>).

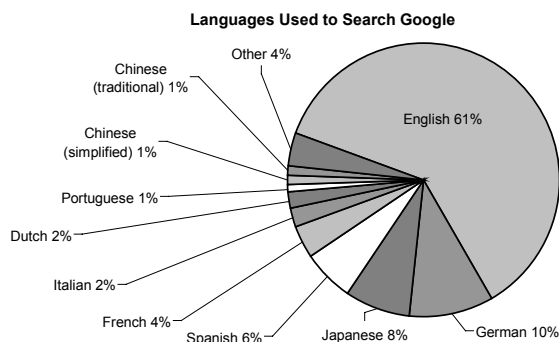


Figure 1: Languages used for searches with the Google web search engine in 2001 October (Google Inc., 2001c).

As Figure 1 shows, more than a third of the web searches done in 2001 October in the Google web search engine were in languages other than English².

This current trend emphasises the use of web engines for searches in various languages and again puts this work into perspective. In Europe alone it is often the case that each country has its own set of official languages and possibly even other regional languages, as it is the case of Spain (Basque, Castilian – commonly referred to as Spanish –, Catalan and Galician), Switzerland (French, German, Italian and Romansh) or the United Kingdom (English, Gaelic and Welsh). Consequently, should a query be done in one of these particular languages, a monolingual web search engine is bound to limit the query to documents written only in the query language.

This paper proposes using multilingual terminology databanks in order to model the contents of multilingual documents and, thus, to provide multilingual access. It describes a method to build multilingual terminology databanks from parallel texts in order to provide an extra multilingual layer for web search and enable searches in documents written in several languages.

² The same trend had also been noticed earlier in 2001 August (Google Inc., 2001b), before the 2001 September 11 attacks in New York, USA, which could have biased the results since more users would be choosing English to search, for example, 'World Trade Centre' or 'Anthrax'.

This paper is structured as follows: the next section gives an overview of what parallel texts are and how they can be aligned. Section 3 describes how to build multilingual terminology databanks from the aligned parallel texts. Section 4 describes how the multilingual databanks can be used for multilingual mining and presents several methodologies that have been proposed. Finally, section 5 presents some conclusions and section 6 draws some future work.

2. Aligning Parallel Texts

In this section we will describe several approaches to parallel texts alignment. First, we start by describing what parallel texts are. Section 2.2 and 2.3 present previous sentence and word alignment methodologies. Section 2.4 describes the alignment method we used.

2.1. Parallel Texts

Parallel texts are sets of texts which are translations of each other in different languages, like the proceedings of the Canadian Parliament – the Canadian Hansards –, which are published in both English and French, or the Official Journal of the European Communities published in the eleven official languages of the European Union³. They have proven to be rich linguistic resources for multilingual text processing and they have become available in a wide range of languages,

However, before it is possible to use them to identify translations of multilingual terms, parallel texts must be aligned first. Text alignment aims at establishing correspondences between parallel texts automatically, either between paragraphs, sentences, or even at sub-sentential level between text segments, phrases, words or sequences of characters.

There have been mainly two approaches to alignment of parallel texts: *sentence alignment* establishes correspondences between sentences only and *word alignment* tries to go a bit deeper into sub-sentential level by establishing correspondences between text segments or words.

2.2. Previous Sentence Alignment Techniques

Back in the early 1990s, sentences were considered as the basic units for alignment. Texts were split into sequences of sentences and alignment algorithms would attempt at making correspondences between the sentences in the parallel texts.

Kay and Röscheisen (1993) were the first to propose an alignment methodology. They assumed that for two sentences written in different languages to correspond, the words in them must also correspond. Their algorithm started by suggesting a tentative alignment of sentences by aligning the first and the last ones of each parallel text. Then, equivalent words were used to align the others. Two words were considered equivalent if they tended to co-occur in the same tentatively aligned sentences. A measure of similarity was computed and if it scored higher than a specific value, it would mean those words were indeed translations. Finally, sentences were aligned if the

number of words associating them was greater than an empirically defined threshold.

In other alternative approaches, less knowledge based, sentences were aligned if they had a proportional number of words (Brown *et al.*, 1991) or characters (Gale and Church, 1991). Each of these authors started from the fact that long sentences tend to have long translations and, conversely, short sentences tend to have short translations. This correlation was the basis for their statistical models. Their algorithms would group sequences of sentences till they had proportional sizes

Brown *et al.* (1991, p. 175) remarked that the error rate was slightly reduced from 3.2% to 2.3% when using some linguistic knowledge like time stamps, question numbers and author names found in the parallel texts. This confirmed the fact that it was sufficient to look at sentence lengths in order to align sentences. Extra linguistic knowledge did not improve the results significantly.

Simard *et al.* (1992) proposed a sentence alignment algorithm which would first align text segments based on the length-based algorithm suggested by Gale and Church (1991), and, if it did not produce a ‘clear’ single best alignment of two text segments, it would proceed into a second pass counting the number of *cognates* shared between them.

According to the Longman Dictionary of Applied-Linguistics, a *cognate* is “a word in one language which is similar in form and meaning to a word in another language because both languages are related” (Richards *et al.*, 1985, p. 43). For example, the words *Parliament* and *Parlement*, in English and French respectively, are cognates. However, if two words have the same or similar forms in two languages but different meanings, they are called false cognates or false friends (Richards *et al.*, 1985, p. 103). For example, the English word *library* and the French word *librairie* are an example of false cognates (Melamed, 1999, p. 114): *library* is translated as *bibliothèque* in French and, conversely, *librairie* as *bookstore* in English.

Simard *et al.* (1992) used a simple rule to test if two words were cognates by checking whether their first four characters were identical (Simard *et al.*, 1992, p. 71), as in *Parliament* and *Parlement*. This simple heuristic proved to be quite useful, providing a great number of lexical cues for alignment though it has some shortcomings. According to it, the English word *government* and the French word *gouvernement* are not cognates. Also, *conservative* and *conseil* (‘council’), in English and French respectively, are wrongly considered as cognates (Melamed, 1999, p. 113). The rule is sensitive to variations in the first four letters but it does not distinguish different word endings.

In order to align English and Chinese sentences, Wu (1994) also used the method based on proportional lengths. He also began by applying a method similar to the one used by Gale and Church (1991) and reported results not much worse than those expected by this algorithm. Still, he claimed sentence alignment precision over 96% when the method incorporated a seed bilingual lexicon of words commonly found in the texts to be aligned (e.g. names of months, like *December* and its equivalent in Chinese 十二月). So, again Wu’s work confirmed that the use of lexical cues would be beneficial for alignment.

The problem with the alignment algorithms which rely solely on sentence sizes is that they tend to break down

³ Danish (da), Dutch (nl), English (en), Finnish (fi), French (fr), German (de), Greek (el), Italian (it), Portuguese (pt), Spanish (es) and Swedish (sv).

when sentence boundaries are not clearly marked in the parallel texts. Sentences need to be clearly identified which means taking the most advantage of the cues provided by full stops. Full stops have to be clearly interpreted in order to check whether they mark a sentence boundary. However, that is not always the case.

Gale and Church (1991, p.179) reported that only 53% of the full stops found in the Wall Street Journal were used to mark sentence boundaries. Full stops may be part of abbreviations (*Dr. A. Bromley*), numbers (1.3%), they are not usually found in headlines (*Tyre production*), they may not even exist because they were not added, or they were either lost or mistaken for noise in the early days when electronic versions of parallel texts were still rare and texts needed to be scanned.

2.3. Previous Word Alignment Techniques

Word alignment is much more fine-grained than sentence alignment since it is no longer done just at sentence level but at word level. Aligned text segments are shorter and, thus, it becomes easier to establish correspondences. However, in contrast with sentence alignment algorithms which permit a margin of tolerance for occasional wrong word matches since sentences generally have many words, they are no longer ‘safety nets’ for word level alignment. Consequently, the penalty on wrong word matches becomes higher and achieving a high precision becomes harder. Should word alignment be the goal, the alignment algorithm must be more ‘careful’ in order to avoid wrong word matches.

Church (1993) showed that by adding some lexical information, alignment of parallel text segments was possible without requiring sentence delimiters. He exploited the notion of orthographic cognates proposed earlier by Simard *et al.* (1992). He used a similar rule: use equal 4-grams in order to find ‘cognate’ (similar) sequences of characters in the parallel texts, i.e. sequences of four characters which are equal in the texts. The method built a graph where a dot at co-ordinates (x, y) meant that there was a match between the 4-grams in positions x and y of both texts. The reliable dots were filtered using an empirically estimated search space.

Fung and Church (1994) dropped the requirement for clear sentence boundaries on a case-study for English-Chinese. It was also the first time alignment procedures were being tested on texts between non-Latin languages and without finding sentence boundaries. Each parallel text was split into K pieces and word correspondences were identified by analysing their distribution across those pieces. In particular, a binary vector of occurrences with size K (hence, the K -vec) would record the occurrence of a word in each of the pieces. Should the word occur in the i -th piece of the text, then the i -th position of the vector would be set to ‘1’. Next, the K -vecs of English and Chinese words were compared in order to find whether two words corresponded. In this way, it was possible to build a rough estimate of a bilingual lexicon to feed the algorithm of Church (1993). In this case, dots would be drawn in the graph each time two translations occurred.

This method was extended in Fung and McKeown (1994). It was also based on the extraction of a small bilingual dictionary based on words with *similar distributions* in the parallel texts. However, instead of K -vecs, which stored the occurrences of words in each of the

K pieces of a text, Fung and McKeown (1994) used vectors that stored the distances between consecutive occurrences of a word (DK-vec’s). For example, if a word appeared at offsets (2380, 2390, 2463, 2565, ...), then the corresponding distances vector would be (10, 73, 102, ...). Should an English word and a Chinese word have similar distance vectors, then they would be used as potential cues for alignment.

In Simard and Plamondon (1998), sentences were aligned using ‘isolated’ cognates as anchors, i.e. cognates that were not mistaken for other cognates within a text window whose width was set to 30% of the text size. Yet, the alignment algorithm would start by aligning words. Each occurrence of a cognate became a dot in a graph according to its offset in each of the parallel texts. Some of those points were filtered if they lied outside an empirically defined search space which would mean they were “not in line” with their neighbouring points. The heuristic values used were found empirically so as to provide the best results and make the best selection of the good alignment cues.

Melamed (1999) also used orthographic cognates. His algorithm filtered noisy correspondence points, i.e. points which were not reliable, according to several heuristics which helped define what a good anchor was. In order to measure word similarity, he defined the ratio of the Longest Common Sub-sequence of characters as follows:

$$Ratio(w_1, w_2) = \frac{Length(Longest\ Common\ Sub\ -\ Sequence(w_1, w_2))}{Max(Length(w_1), Length(w_2))}$$

where w_1 and w_2 are the two words to be compared (Melamed, 1999, p.113). This measure compares the length of the longest common sub-sequence of characters with the length of the longest token. For example, for *government* and *gouvernement*, the ratio is 10 (the length of *government*) over 12 (the length of *gouvernement*) whereas the ratio is just 6 over 12 for *conservative* and *conseil* (‘council’). This measure tends to favour long sequences similar to the longest word and to penalise sequences which are too short compared to a long word. So, for this very reason, it fails to consider *gouvernement* and *governo* in French and Portuguese as cognates because *governo* is shorter. Their ratio is also 6 over 12.

For alignment purposes, Melamed (1999) selects all pairs of words which have a ratio above a certain threshold, empirically selected. Still, this comparison measure seems to provide better results than the one first proposed by Simard *et al.* (1992) but it is not also based on a statistically supported study.

2.4. The Alignment Methodology

In contrast with the previous approaches, Ribeiro *et al.* (2000) present a statistically supported method for word alignment of parallel texts which does not require either clearly delimited sentences or previous linguistic knowledge of the texts languages. It was applied to parallel texts in the 11 official languages of the European Union and also to parallel texts in Portuguese and Chinese (Ribeiro *et al.*, 2001a).

In particular, the alignment methodology selects alignment points using filters based on linear regression lines properties. The points are generated from the offsets of lexical cues provided by equal tokens (like numbers, proper names, punctuation marks) which occur with the

same frequency within a parallel text segment. Since the algorithm is recursive, even if some token happens not to have the same ‘global’ frequency, it may end up being used as an alignment point in a ‘local’ analysis of smaller text segments.

This algorithm was later extended in Ribeiro *et al.*, (2001b) to handle typical sequences of characters common to a particular pair of languages. Instead of using heuristics to identify cognate words or of using particular sizes of n -grams of characters to find similar sequences of characters, they made statistical data analyses of contiguous and non-contiguous sequences of characters to extract associated character units from each pair of languages. They were able to find typical sequences of characters in the beginning of words, such as •Comis, for *Comissão* and *Comisión* (‘Commission’) in Portuguese and Spanish, in the middle of words, as in *f_rma* which matches both *information* and *informação* in English and Portuguese respectively, or across word boundaries, as *i_re•ci* for the Portuguese–French pair as in *livre•circulação* and *libre•circulation* (‘free movement’).

The average alignment precision is over 90% for aligned parallel texts in Portuguese with all the other official languages of the European Union. This is the precision of a word alignment algorithm which, in contrast with other algorithms, does not rely on language specific knowledge, lists of stop words to avoid noise generated by frequent words or extra seed bilingual lexicons.

3. Building Multilingual Databanks

Aligned parallel texts are ideal sources to extract Translation Equivalents for they provide the correspondences between the original text and their translations in other languages. They allow easily the examination of the way specific words or terms are translated into other languages. Consequently, they can reduce the amount of effort necessary to build Translation Databanks.

For this experiment we used a sample of parallel texts from three sources: records of the Written Questions to the European Commission (ELRA, 1997), records of Debates in the European Parliament (ELRA, 1997) and Judgements of The Court of Justice of the European Communities in all the languages of the European Union.

In order to identify relevant multiword units, it has been common practice to do it by hand coding regular syntactic patterns, like the sequence ‘Noun Noun’ (e.g. *Web Mining*). Finite state automata are then used to recognise typical sequences of words in the texts which comply with these patterns. For example, Daille (1995) used several syntactic patterns to identify terms with two words such that they were either two nouns or a noun and an adjective, as in *liaison par satellite* (‘satellite link’) or *station terrienne* (‘earth station’). Fung and McKeown (1997) also used specific syntactic patterns to extract multiword terms in order to compile a list of reliable pairs of translations for a further extension to their previous alignment algorithms (Fung and McKeown, 1994).

Although terms are generally covered by some characteristic patterns, this work has not started from a particular set of patterns so as not to constrain the structure of the multiword units.

In order to build the Multilingual Terminology Databank, we extracted terms from the parallel texts using

a methodology described in da Silva *et al.* (1999). This methodology is based on the idea that the more cohesive a group of n words is, the higher its cohesiveness score. The algorithm assumes that the score of a good multiword unit must be a local maximum, i.e. the cohesion of the set of n words is higher than any subset of $n-1$ words contained in it and higher than the cohesion of any superset of $n+1$ words which contains it. Thus, the algorithm is able to select, for example, *common rules and standards* as a relevant multiword term but not *common rules and* or *common rules and standards for*, because the scores of these multiword units are lower. The figure below shows some extracted terms:

English	French	Portuguese
<i>combined</i>	<i>autorités</i>	<i>autoridades</i>
<i>nomenclature</i>	<i>douanières</i>	<i>aduaneiras</i>
<i>customs</i>	<i>États membres</i>	<i>Estados –</i>
<i>authorities</i>	<i>matières</i>	<i>Membros</i>
<i>intervention</i>	<i>nucléaires</i>	<i>materiais</i>
<i>agency</i>	<i>nomenclature</i>	<i>nucleares</i>
<i>Member States</i>	<i>combinée</i>	<i>Nomenclatura</i>
<i>nuclear material</i>	<i>organisme</i>	<i>Combinada</i>
	<i>d’intervention</i>	<i>organismo de</i>
		<i>intervenção</i>

Table 1: A sample of extracted multiword terms in English, French and Portuguese.

The methodology has proven to be quite adequate to be used across several languages. In this way, we were able to capture multiword terms for each language and build databanks of terms. However, it still remains to be seen how the relations between them can be established, i.e. how to build the multilingual terminology databank of equivalent translations.

The key issue in the extraction of Translation Equivalents is to find a correlation between co-occurrences of terms in the aligned parallel texts. In general, if two terms co-occur often in aligned text segments, then they are likely to be *equivalent*.

The alignment of parallel texts splits them into small aligned text segments and reduces the number of words / terms that must be checked for co-occurrence in each parallel text segment. The shorter the segments, the better. In order to identify Translation Equivalents, the *distribution similarity* of words / terms must be analysed in the aligned segments.

Following the conventional information retrieval methodology (Salton and McGill, 1983), the information on the occurrence of words (or terms) is usually represented in vector forms. For example, if a word w occurs in segments 1, 2 and 5 out of a total of five segments, then the following *occurrences vector* is built: $w = (1, 1, 0, 0, 1)$. In this binary vector, each ‘0’ and ‘1’ represents the absence and presence of the word w in each of the five segments.

In this way, a set of occurrence vectors can be built for each of the terms found. Next, for each pair of source and target terms a *co-occurrence vector* is built where the i -th position of the vector is set to ‘1’ if both terms occur in the i -th aligned text segment. Next, a *contingency table* is built for each pair of source–target terms by counting the number of ‘0’s and ‘1’s in the occurrences vectors.

$n: 162347$	Επιτροπή των Ευρωπαϊκών Κοινοτήτων	× Επιτροπή των Ευρωπαϊκών Κοινοτήτων
Comissão das Comunidades Europeias	$a: 499$	$b: 102$
× Comissão das Comunidades Europeias	$c: 96$	$d: 161650$

Table 2: Contingency table for the pair *Επιτροπή των Ευρωπαϊκών Κοινοτήτων* (Epitropé tos Europaikós Koinotétos) and *Comissão das Comunidades Europeias* (‘Commission of the European Communities’).

This table stores the *number of aligned segments* that contain:

- a : both terms;
- b : the Portuguese term but not the Greek term;
- c : the Greek term but not the Portuguese term; and,
- d : neither of those terms.

These amounts can be computed from the occurrences vectors as follows:

- n : the size of the occurrences vectors;
- a : the number of ‘1’s in the co-occurrence vector;
- b : the number of ‘1’s found in the Portuguese term occurrences vector minus a ;
- c : the number of ‘1’s found in the Greek term occurrences vector minus a ; and,
- d : $n - a - b - c$.

The difference between the total number of occurrences of both words may result either from different translations made by the translators themselves and / or from some occasional misalignment. Different translations may be due to syntactic constraints or to alternative translations the human translator decided to make.

Several measures of similarity have been proposed to use the information in the contingency tables in order to analyse the similarity of words and identify Translation Equivalents. We have used the Average Mutual Information as this similarity measure has proven to be appropriate for the task of identifying Translation Equivalents. The Average Mutual Information is computed as follows:

$$I(X;Y) = \sum_{x=\{0,1\}} \sum_{y=\{0,1\}} p(X=x, Y=y) \log_2 \left(\frac{p(X=x, Y=y)}{p(X=x)p(Y=y)} \right)$$

where X and Y are the two terms to be tested as translations. This formula is in contrast with the Specific

Mutual Information, which is quite sensitive to rare co-occurrences, and which corresponds only to the last term of the sum. In this formula, $p(x=1, y=0)$ is the probability that term X occurs but term Y does not. Figure 2 shows some Translation Equivalents extracted in English, Greek and Portuguese.

Since we have used a general purpose terminology extractor, it extracts not only domain specific terms but also general language patterns. This happens because it tends to capture typical sequences of tokens independently of whether they are domain specific or not. The extractor was not developed to identify domain specific terminology though it is able to extract it too. We believe that by clustering documents and feeding those clusters of documents independently to the extractor it will be possible to distinguish domain specific terms from general language patterns. da Silva *et al.* (2001) proposes an unsupervised and language independent method to cluster documents to be used for this task.

Finally, by *re-feeding* the extracted Translation Equivalents back into the aligner it is possible to increase the number of potential anchors and, consequently, the number of new lexical cues available for the generation of correspondence points. The more correspondence points, the more fine-grained the alignment can be and the better the extracted equivalents can be. This means that alignment precision may improve. This is especially important for pairs of languages which share few lexical cues which can be used for alignment (like Portuguese and Chinese, as an extreme case).

4. Mining Multilingual Documents

Mining multilingual documents is a generalisation of the problem of mining documents that contain expressions which do not match exactly the ones in the query text. Fluhr (1995) made a survey of several approaches used for Multilingual Information Retrieval.

One traditional approach consists of using a *controlled vocabulary* both to index and retrieve documents, like the one used by Reuters or the Eurovoc (1995). Each document is indexed with a set of descriptors and queries are performed using this set of keywords. Queries are reformulated into the other languages by looking up the translations of the descriptors in a multilingual databank which contains the translations of each descriptor in the other languages.

English	Greek	Portuguese
JUDGMENT OF THE COURT	ΑΠΟΦΑΣΗ ΤΟΥ ΔΙΚΑΣΤΗΡΙΟΥ	ACÓRDÃO DO TRIBUNAL DE JUSTIÇA
Advocate General	γενικός εισαγγελέας	advogado – geral
Language of the case	Γλώσσα διαδικασίας	Língua do processo
Commission of the European Communities	Επιτροπή των Ευρωπαϊκών Κοινοτήτων	Comissão das Comunidades Europeias
Member States	κρατών μελών	Estados – membros
Act of Accession	Πράξεως Προσχωρήσεως	Acto de adesão
President of the Chamber	πρόεδρος τμήματος	presidente de secção
First Chamber	πρώτο τμήμα	Primeira Secção

Figure 2: A sample of Translation Equivalents obtained from the aligned texts in English, Greek and Portuguese.

However, the problem with this approach is that it limits queries to using the set of descriptors available instead of using full text words. An alternative method builds a matrix which links full text words to the set of controlled descriptors. This matrix can be built either manually or automatically by *learning* from previously indexed texts – a *text categorisation* task (Yang 1999). Once a query is posted, this matrix is looked up in order to find which descriptors are more associated with the words in the query. Finally, the translations of the descriptors are looked up in the multilingual databank in order to reformulate the query in the other languages.

Nevertheless, the use of controlled languages means that queries are somehow limited to the set of descriptors available. Alternative approaches can either translate the query – *query reformulation through translation* – or even the whole set of documents. Although there is some debate on the benefits and disadvantages of each one, reformulating the query through translation seems to be the simplest strategy since the latter option requires translating each document into all the other languages, which does not scale up well. Still, a query translated with errors may yield disappointing results if it has unresolved lexical ambiguities.

Anyhow, should parallel texts be available, they can become quite helpful. Some approaches exploit this fact by retrieving not only the documents most similar to the query posted but also their parallel versions. Then, the parallel texts can be used as a secondary query to retrieve similar untranslated documents in the other languages and even more parallel documents in the original query language should they be available.

In order for the search engine to retrieve the documents most similar to a query, several approaches have been suggested though most of them are based on the Vector Space model (Salton and McGill, 1983). In this model, documents are represented in a n -dimensional space, where n is the number of different words found in the texts – the term-document matrix. Both queries and documents are represented with n -dimensional vectors of term weights. Usually, terms are weighted using TF \times IDF, the term frequency \times the inverse document frequency of a term, i.e. the inverse of the number of documents in which the term occurs. Then, a document is considered relevant for a query if the query and the document vectors are *similar*. The similarity of two vectors can be computed using the cosine measure.⁴

In contrast with the previous model, the Generalised Vector Space Model (Wong *et al.*, 1985) takes into account the fact that terms are correlated. The assumption of this model is that two words are semantically similar if they tend to occur in the same documents, i.e. have similar document vectors in the term-document matrix. This Generalised model bears this in mind.

In the Pseudo-Relevance Feedback model, the initial query is expanded by adding to it terms found in the first set of retrieved documents, assuming that the top ranking documents are indeed relevant. This new extended query is posted again to the search engine in order to retrieve more documents (Salton and Buckley, 1990). Should there be parallel versions available for these documents, they can be used instead. This is the extension of the

monolingual Pseudo-Relevance Feedback approach to multilingual retrieval suggested by Carbonell *et al.* (1997).

The Latent Semantic Indexing model (Deerwester *et al.*, 1990) is the next step after the previous model. It is also sensitive to co-occurrences of terms in the same document when it computes the similarity between the query and each document. The whole set of documents is *reduced* so that a smaller set is more representative for the content of the documents. This model was adapted to multilingual retrieval by Dumais *et al.* (1996), using parallel texts for training.

As for the approaches which expand the query and reformulate it with a translation, Machine Translation systems would probably be a good option and be able to provide good translations if queries were usually formulated as sentences or paragraphs. However, queries tend to be short and users tend to give isolated words for which Machine Translation systems performance degrades. Some alternative strategies have been suggested:

- look up each query word in a bilingual dictionary and use all possible translations;
- use a sentence aligned corpus and expand the query using every sentence in which all the query words co-occur; and,
- use an aligned corpus to build a translation databank.

The work presented in this paper fits in this last alternative. In this case, it becomes important to have good multilingual terminology databanks; otherwise, the reformulation of the query through the translation may not be correct if terms are not properly identified and translated. Carbonell *et al.* (1997) made an evaluation of several multilingual retrieval methods and concluded that query expansion by translation using a corpus-based ‘translation matrix’ provided the best results even when compared with a general purpose dictionary. Another reason why this approach seems to be better than a Machine Translation system is that it is easier to build a databank of translations for a new language, given parallel texts are available, than it is to build a Machine Translation system for the new language.

Thus, rather than translating each of the query words individually and providing all their possible translations, or using all sentences in which the words occur, the multilingual databank provides a simple means to make accurate translations of terms and, consequently, reduce their ambiguity. Furthermore, there are times when not even combining each of the possible word translations individually provides a possible compound translation, like *border crossing point* and *poste frontière* (‘border post’) in French, *hang gliding* and *asa delta* (‘delta wing’) in Portuguese, or even the common English phrasal verbs like *put up with* whose word for word translation are hardly combinable for other languages.

Thus, once a query is posted to a search engine, the multilingual databank of terminology can be used to translate the query terms into the available languages and posting subsequently monolingual searches in order to find relevant documents in the other languages. Had each of the query words been translated word for word, all alternative translations would have to be used which may lead to a long list of possible translations combinations. This increases the search space as more documents are bound to contain each of the words individually rather

⁴ For a simple introduction, see, for example, Manning and Schütze (1999).

than the full correct translation. Also, a word for word translation may lead to no valid translation at all as shown with the examples above.

5. Conclusions

Currently, web search engines hardly support multilingual retrieval. Should a query be made in a language for which no relevant document exists, it will be unsuccessful. In the near future, it should be possible to access information independently of the language of the user and independently of the language in which the source text is written. This is what multilingual retrieval promises.

In this paper, we have made a small contribution to it. We have focussed on building multilingual terminology databanks from aligned texts in order to use them for multilingual retrieval. Compound words are particularly important in technical fields where their translation cannot be usually done word for word.

This paper has presented a methodology to extract terminology Translation Equivalents from aligned parallel texts so as to add a multilingual layer to search engines and allow multilingual searches by query expansion through translation to the other languages. In particular, this paper has described language independent and statistically supported methodologies to align parallel texts, extract multiword terms and find translation equivalents in the aligned texts. None of the techniques used assumes any language specific knowledge nor requires human hand coding of linguistic information.

We believe that by providing a multilingual terminology databank to multilingual search engines, it becomes possible to make reliable multilingual searches of compound terms as attested by Carbonell *et al.* (1997). Instead of building a databank of word translations in several languages, this work reports on the generation of a multilingual databank of multiword units from parallel texts. This makes translation of queries which contain compound terms less liable to errors. Also it reduces the search space of documents since terms can be identified in the query and translated as a unit instead of translating each of the words individually and retrieving documents which contain any of the possible translations.

6. Future Work

We need to make comparative evaluations on the retrieval performance on multilingual retrieval systems enhanced by the multilingual databank extracted from the parallel texts. It would also be interesting to check whether sub-sentential aligned text segments might be of some help for the translation of queries when the translation databank is not able to provide a translation. This strategy would simultaneously combine a query expansion approach based on a databank of translations with a query expansion approach based on aligned sub-sentential text segments.

As for the terminology extractor, its input needs to be normalised in order to avoid word variants. This will improve the accuracy of the translation equivalents extracted. In addition, it will increase the number of extracted translations by eliminating the sparse data problem due to alternative word variants. There are problems with highly inflectional languages like Greek, Finnish or even Portuguese. For example, the English

adjective *public* can be translated into Portuguese as *público*, *pública*, *públicos*, or *públicas*, depending on the gender and number of the noun it qualifies. As a result, Translation Equivalents of terms which suffer variants tend to have low scores. We believe we can accomplish this task, also using language independent methods, by extracting typical sequences of characters using a methodology similar to the one used to extract typical sequences of characters in the alignment algorithm reported in Ribeiro *et al.* (2001b). This would identify common sequences of characters for inflected words though it may become harder for words which suffer radical changes when inflected. The verb *to be* is an English extreme case where a single word has eight variants: *be / being / am / are / is / was / were / been*.

Last but not the least, we need to distinguish domain specific terms from general language patterns. For this, we will use a method proposed by da Silva *et al.* (2001) to cluster documents according to domain specificity.

7. Acknowledgements

This work was partly funded by project Tradaut-pt.

8. References

- Brown, P., Lai, J. and Mercer, R. (1991). Aligning Sentences in Parallel Corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics* (pp. 169–176). Berkeley, California, USA.
- Brown, R., Carbonell, J. and Yang, Y. (2000). Automatic Dictionary Extraction for Cross-Language Information Retrieval. In J. Véronis (ed.), *Parallel Text Processing: Alignment and Use of Translation Corpora* (pp. 275–298). Dordrecht, The Netherlands: Kluwer Academic Publisher.
- Carbonell, J., Yang, Y., Frederking, R., Brown, R., Geng, Y. and Lee, D. (1997). Translingual Information Retrieval: A Comparative Evaluation. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence – IJCAI 97* (pp. 708–715). Volume I. Nagoya, Japan.
- Church, K. (1993). Char_align: A Program for Aligning Parallel Texts at the Character Level. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics* (pp. 1–8). Columbus, Ohio, USA.
- Daille, B. (1995). Combined Approach for Terminology Extraction: Lexical Statistics and Linguistic Filtering. In *UCREL (University Centre for Computer Corpus Research on Language) Technical Papers. 5*. Lancaster, United Kingdom: University of Lancaster, Department of Linguistics.
- Deerwester, S., Dumais, S., Furnas, G., Landauer, T. and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41 (6), 391–498.
- Dumais, S., Landauer T. and Littman, M. (1996). Automatic Cross-Linguistic Information Retrieval Using Latent Semantic Indexing. In *Proceedings of the 19th Annual International ACM (Association for Computing Machinery) – SIGIR (Special Interest Group in Information Retrieval) Conference on Research and Development in Information Retrieval – SIGIR '96* (pp. 16–23). Zurich, Switzerland.

- ELRA – European Language Resources Association (1997). *Multilingual Corpora for Co-operation*, Disk 2 of 2. Paris, France, 454 MBytes.
- Eurovoc (1995). *Thesaurus Eurovoc – Volume 2: Subject-Oriented Version*. Annex to the index of the Official Journal of the EC. Luxembourg, Office for Official Publications of the European Communities. Retrieved from <http://europa.eu.int/celex/eurovoc> on Mon, 2002 Apr 15.
- Fluhr, C. (1995). Multilingual Information Retrieval. In R. Cole, J. Mariani, H. Uszkoreit, A. Zaenen and V. Zue (eds.), *Survey of the State of the Art in Human Language Technology*. Retrieved from <http://cslu.cse.ogi.edu/HLTsurvey/Ch8Node7.html#Section85> on Mon, 2002 Apr 15.
- Fung, P. and Church, K. (1994). K-vec: A New Approach for Aligning Parallel Texts. In *Proceedings of the 15th International Conference on Computational Linguistics – Coling’94* (pp. 1096–1102), Kyoto, Japan.
- Fung, P. and McKeown, K. (1994). Aligning Noisy Parallel Corpora across Language Groups: Word Pair Feature Matching by Dynamic Time Warping. In *Technology Partnerships for Crossing the Language Barrier: Proceedings of the First Conference of the Association for Machine Translation in the Americas* (pp. 81–88). Columbia, Maryland, USA.
- Fung, P. and McKeown, K. (1997). A Technical Word-and Term-Translation Aid Using Noisy Parallel Corpora across Language Groups. *Machine Translation*, 12 (1–2), 53–87. Dordrecht, The Netherlands: Kluwer Academic Publisher.
- Gale, W. and Church, K. (1991). A Program for Aligning Sentences in Bilingual Corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics* (pp. 177–184). Berkeley, California, USA (short version). Also (1993) *Computational Linguistics*, 19 (1), 75–102 (long version).
- Google Inc. (2001a). *Search 3 Billion Documents Using Google*. Retrieved from <http://www.google.com/3.html> on Tue, 2001 December 11.
- Google Inc. (2001c). *Google Zeitgeist – Search Patterns, Trends, and Surprises According to Google*. Google Press Centre: Zeitgeist, 2001 October issue. Retrieved from <http://www.google.com/press/zeitgeist/zeitgeist-oct.html> on Wed, 2002 January 9.
- Google Inc. (2001b). *Google Zeitgeist – Search Patterns, Trends, and Surprises According to Google*. Google Press Centre: Zeitgeist, 2001 August issue. Retrieved from <http://www.google.com/press/zeitgeist/zeitgeist-aug.html> on Wed, 2002 January 9.
- Kay, M. and Röscheisen, M. (1993). Text-Translation Alignment. *Computational Linguistics*, 19 (1), 121–142.
- Landauer, T. and Littman, M. (1990). Fully-Automatic Cross-Language Document Retrieval Using Latent Semantic Indexing. In *Proceedings of the 6th Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary* (pp. 40–62). Waterloo, Canada.
- Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing* (680 p.). 4th edition, Cambridge, Massachusetts, USA: The MIT Press.
- Melamed, I. (1999). Bitext Maps and Alignment via Pattern Recognition. *Computational Linguistics*, 25 (1), 107–130.
- Ribeiro, A., Lopes, G. and Mexia, J. (2000). Using Confidence Bands for Parallel Texts Alignment. In *Proceedings of the 38th Conference of the Association for Computational Linguistics* (pp. 432–439). Hong Kong, China.
- Ribeiro, A., Lopes, G. and Mexia, J. (2001a). Extracting Translation Equivalents from Portuguese-Chinese Parallel Texts. *Journal of Studies in Lexicography*, 11 (1), 118–194. Seoul, South Korea: Yonsei University.
- Ribeiro, A., Dias, G., Lopes, G. and Mexia, J. (2001b). Cognates Alignment. In B. Maegaard (ed.), *Proceedings of the Machine Translation Summit VIII – MT Summit VIII – Machine Translation in the Information Age* (pp. 287–292). Santiago de Compostela, Spain.
- Richards, J., Platt, J. and Weber, H. (1985). *Longman Dictionary of Applied Linguistics*. London, United Kingdom: Longman.
- Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval*. New York, USA: McGraw-Hill (448 p.).
- Salton G. and Buckley, C. (1990). Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 41 (4), 182–188.
- da Silva, J., Dias, G., Guilloiré, S. and Lopes, J. (1999). Using Localmaxs Algorithms for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. In P. Barahona and J. Alferes (eds.), *Progress in Artificial Intelligence – Lecture Notes in Artificial Intelligence*, 1695 (pp. 113–132). Berlin, Germany: Springer-Verlag.
- da Silva, J., Mexia, J., Coelho, C. and Lopes, J. (2001). Document Clustering and Cluster Topic Extraction in Multilingual Corpora. In N. Cercone, T. Lin and X. Wu (eds.), *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 2001 International Conference on Data Mining – ICDM’01* (pp. 513–520). San Jose, California, USA: IEEE Computer Society Press.
- Simard, M. and Plamondon, P. (1998). Bilingual Sentence Alignment: Balancing Robustness and Accuracy. *Machine Translation*, 13 (1), 59–80. Dordrecht, The Netherlands: Kluwer Academic Publisher.
- Simard, M., Foster, G. and Isabelle, P. (1992). Using Cognates to Align Sentences in Bilingual Corpora. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation TMI-92* (pp. 67–81). Montréal, Canada.
- Wu, D. (1994). Aligning a Parallel English–Chinese Corpus Statistically with Lexical Criteria. In *Proceedings of the 32nd Annual Conference of the Association for Computational Linguistics* (pp. 80–87), Las Cruces, New Mexico, USA.
- Wong, S., Ziarko, W. and Wong, P. (1985). Generalized Vector Space Model in Information Retrieval. In *Proceedings of the 8th Annual International ACM (Association for Computing Machinery) – SIGIR (Special Interest Group in Information Retrieval) Conference on Research and Development in Information Retrieval – SIGIR’85* (pp. 18–25). New York, USA.
- Yang, Y. (1999). An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, 1, 69–90.

Grammar Learning by Partition Search

Anja Belz

ITRI
University of Brighton
Lewes Road
Brighton BN2 4GJ, UK
Anja.Belz@itri.brighton.ac.uk

Abstract

This paper describes Grammar Learning by Partition Search, a general method for automatically constructing grammars for a range of parsing tasks. Given a base grammar, a training corpus, and a parsing task, Partition Search constructs an optimised probabilistic context-free grammar by searching a space of nonterminal set partitions, looking for a partition that maximises parsing performance and minimises grammar size. The method can be used to optimise grammars in terms of size and performance, or to adapt existing grammars to new parsing tasks and new domains. This paper reports an example application to optimising a base grammar extracted from the Wall Street Journal Corpus. Partition Search improves parsing performance by up to 5.29%, and reduces grammar size by up to 16.89%. Parsing results are better than in existing treebank grammar research, and compared to other grammar compression methods, Partition Search has the advantage of achieving compression without loss of grammar coverage.

1. Introduction

Grammar Learning by Partition Search is a computational learning method that constructs probabilistic grammars optimised for a given domain or parsing task. The main idea behind this method is that new grammars can be derived from existing ones by simple operations on nonterminal sets. Automatically carrying out different combinations of such operations and testing the derived grammars' size and performance makes it possible to automatically optimise the grammars.

The main practical applications of Grammar Learning by Partition Search are the optimisation of an existing grammar's size and performance, and the adaptation of existing grammars to new tasks. Results for optimising a base grammar extracted from the Wall Street Journal Corpus (WSJC) are reported here, while results for adapting the same base grammar to different noun phrase extraction tasks are reported elsewhere (Belz, 2002).

This paper is organised in two main sections. Section 2. describes Grammar Learning by Partition Search. Section 3. reports experiments and results for NP identification and NP chunking.

2. Learning PCFGs by Partition Search

Partition Search Grammar Learning starts from the idea that new context-free grammars (CFGs) can be created from old simply by modifying the nonterminal sets, *merging* and *splitting* subsets of nonterminals. For example, for certain parsing tasks it is useful to *split* a single verb phrase category into verb phrases that are headed by a modal verb and those that are not, whereas for other parsing tasks, the added grammar complexity is avoidable. In another context, it may not be necessary to distinguish noun phrases in subject position from first objects and second objects, making it possible to *merge* the three categories into one.

The usefulness of such split and merge operations can be measured by their effect on a grammar's size (number of rules and nonterminals) and performance (parsing accuracy

on a given task). Grammar Learning by Partition Search automatically tries out different combinations of merge and split operations and therefore can automatically optimise a grammar's size and performance on a given task.

2.1. Preliminary definitions

Definition 1 Set Partition

A partition of a nonempty set A is a subset Π of 2^A such that \emptyset is not an element of Π and each element of A is in one and only one set in Π .

The partition of A where all elements are singleton sets is called the *trivial partition* of A .

Definition 2 Probabilistic Context-Free Grammar¹

A Probabilistic Context-Free Grammar (PCFG) is a 4-tuple (W, N, N_S, R) , where W is a set of terminal symbols, N is a set of nonterminal symbols, $N_S \in N$ is a start symbol, and $R = \{(r_1, p(r_1)), \dots, (r_m, p(r_m))\}$ is a set of rules with associated probabilities. Each rule r_i is of the form $n \rightarrow \alpha$, where n is a nonterminal, and α is a string of terminals and nonterminals. For each nonterminal n , the values of all $p(n \rightarrow \alpha_i)$ sum to one, or: $\sum_{i:(n \rightarrow \alpha_i, p(n \rightarrow \alpha_i)) \in R} p(n \rightarrow \alpha_i) = 1$.

2.2. Generalising and Specialising PCFGs through Nonterminal Set Operations

2.2.1. Nonterminal merging

Consider two PCFGs G and G' :

¹This definition is for PCFGs with a single start symbol, to simplify the definition of PCFG Partitioning below.

$$\begin{aligned}
G &= (W, N, N_S, R), \\
W &= \{ \text{NNS, DET, NN, VBD, JJ} \} \\
N &= \{ S, \text{NP-SUBJ, VP, NP-OBJ} \} \\
N_S &= S \\
R &= \{ (S \rightarrow \text{NP-SUBJ VP, 1}), \\
&\quad (\text{NP-SUBJ} \rightarrow \text{NNS, 0.5}), \\
&\quad (\text{NP-SUBJ} \rightarrow \text{DET NN, 0.5}), \\
&\quad (\text{VP} \rightarrow \text{VBD NP-OBJ, 1}), \\
&\quad (\text{NP-OBJ} \rightarrow \text{NNS, 0.75}), \\
&\quad (\text{NP-OBJ} \rightarrow \text{DET JJ NNS, 0.25}) \}
\end{aligned}$$

$$\begin{aligned}
G' &= (W, N', N_S, R'), \\
W &= \{ \text{NNS, DET, NN, VBD, JJ} \} \\
N' &= \{ S, \text{NP, VP} \} \\
N_S &= S \\
R' &= \{ (S \rightarrow \text{NP VP, 1}), \\
&\quad (\text{NP} \rightarrow \text{NNS, 0.625}), \\
&\quad (\text{NP} \rightarrow \text{DET NN, 0.25}), \\
&\quad (\text{VP} \rightarrow \text{VBD NP, 1}), \\
&\quad (\text{NP} \rightarrow \text{DET JJ NNS, 0.125}) \}
\end{aligned}$$

Intuitively, to derive G' from G , the two nonterminals NP-SUBJ and NP-OBJ are merged into a single new nonterminal NP. This merge results in two rules from R becoming identical in R' : both NP-SUBJ \rightarrow NNS and NP-OBJ \rightarrow NNS become NP \rightarrow NNS. One way of determining the probability of the new rule NP \rightarrow NNS is to sum the probabilities of the old rules and renormalise by the number of nonterminals that are being merged². In the above example therefore $p(\text{NP} \rightarrow \text{NNS}) = (0.5 + 0.75)/2 = 0.625$ ³.

An alternative would be to reestimate the new grammar on some corpus, but this is not appropriate in the current context: merge operations are used in a search process (see below), and it would be expensive to reestimate each new candidate grammar derived by a merge. It is better to use any available training data to estimate the original grammar's probabilities, then the probabilities of all derived grammars can simply be calculated as described above without expensive corpus reestimation.

The new grammar G' derived from an old grammar G by merging nonterminals in G is a generalisation of G : the language of G' , or $L(G')$, is a superset of the language of G , or $L(G)$. E.g., in the above example, `det jj nns vbd det jj nns` is in $L(G')$ but not in $L(G)$. For any sentence $s \in L(G)$, the parses assigned to s by G' form a superset of the set of parses assigned to s by G . The probabilities of parses for s can change, and so can the probability ranking of the parses, i.e. the most likely parse for s under G may be different from the most likely parse for s under G' . Finally, G' has the same number of rules as G or fewer.

2.2.2. Nonterminal splitting

Deriving a new PCFG from an old one by splitting nonterminals in the old PCFG is not quite the exact reverse of deriving a new PCFG by merging nonterminals. The difference lies in determining probabilities for new rules.

²Reestimating the probabilities on the training corpus would of course produce identical results.

³Renormalisation is necessary because the probabilities of all rules expanding the same nonterminal sum to one, therefore the probabilities of all rules expanding a new nonterminal resulting from merging n old nonterminals will sum to n .

Consider the following grammars G and G' :

$$\begin{aligned}
G &= (W, N, N_S, R), \\
W &= \{ \text{NNS, DET, NN, VBD, JJ} \} \\
N &= \{ S, \text{NP, VP} \} \\
N_S &= S \\
R &= \{ (S \rightarrow \text{NP VP, 1}), \\
&\quad (\text{NP} \rightarrow \text{NNS, 0.625}), \\
&\quad (\text{NP} \rightarrow \text{DET NN, 0.25}), \\
&\quad (\text{VP} \rightarrow \text{VBD NP, 1}), \\
&\quad (\text{NP} \rightarrow \text{DET JJ NNS, 0.125}) \}
\end{aligned}$$

$$\begin{aligned}
G' &= (W, N', N_S, R'), \\
W &= \{ \text{NNS, DET, NN, VBD, JJ} \} \\
N' &= \{ S, \text{NP-SUBJ, VP, NP-OBJ} \} \\
N_S &= S \\
R' &= \{ (S \rightarrow \text{NP-SUBJ VP, } \frac{1}{3}), \\
&\quad (S \rightarrow \text{NP-OBJ VP, } \frac{1}{3}), \\
&\quad (\text{NP-SUBJ} \rightarrow \text{NNS, } \frac{1}{3}), \\
&\quad (\text{NP-SUBJ} \rightarrow \text{DET NN, } \frac{1}{3}), \\
&\quad (\text{NP-SUBJ} \rightarrow \text{DET JJ NNS, } \frac{1}{3}), \\
&\quad (\text{VP} \rightarrow \text{VBD NP-SUBJ, } \frac{1}{3}), \\
&\quad (\text{VP} \rightarrow \text{VBD NP-OBJ, } \frac{1}{3}), \\
&\quad (\text{NP-OBJ} \rightarrow \text{NNS, } \frac{1}{3}), \\
&\quad (\text{NP-OBJ} \rightarrow \text{DET NN, } \frac{1}{3}), \\
&\quad (\text{NP-OBJ} \rightarrow \text{DET JJ NNS, } \frac{1}{3}) \}
\end{aligned}$$

To derive G' from G , the single nonterminal NP is split into two nonterminals NP-SUBJ and NP-OBJ. This split results in several new rules. For example, for the old rule NP \rightarrow NNS, there now are two new rules NP-SUBJ \rightarrow NNS and NP-OBJ \rightarrow NNS. One possibility for determining the new rule probabilities is to redistribute the old probability mass evenly among them, i.e. $p(\text{NP} \rightarrow \text{NNS}) = p(\text{NP-SUBJ} \rightarrow \text{NNS}) = p(\text{NP-OBJ} \rightarrow \text{NNS})$. However, then there would be no benefit at all from performing such a split: the resulting grammar would be larger, the most likely parses remain unchanged, and for each parse p under G that contains a nonterminal NT participating in a split, there would be at least two equally likely parses under G' .

The new probabilities cannot be calculated directly from G . The redistribution of the probability mass has to be motivated from a knowledge source outside of G . One way to proceed is to estimate the new rule probabilities on the original corpus — provided that it contains the information on the basis of which a split operation was performed in extractable form. For the current example, a corpus in which objects and subjects are annotated could be used to estimate the probabilities of the rules in G' , and might yield the following rule set R' (which reflects the fact that in English, the NP in a sentence NP VP is (usually) a subject, whereas the NP in a VP consisting of a verb followed by an NP is an object):

$$\begin{aligned}
R' &= \{ (S \rightarrow \text{NP-SUBJ VP, 1}), \\
&\quad (S \rightarrow \text{NP-OBJ VP, 0}), \\
&\quad (\text{NP-SUBJ} \rightarrow \text{NNS, 0.5}), \\
&\quad (\text{NP-SUBJ} \rightarrow \text{DET NN, 0.5}), \\
&\quad (\text{NP-SUBJ} \rightarrow \text{DET JJ NNS, 0}) \} \\
&\quad (\text{VP} \rightarrow \text{VBD NP-SUBJ, 0}), \\
&\quad (\text{VP} \rightarrow \text{VBD NP-OBJ, 1}), \\
&\quad (\text{NP-OBJ} \rightarrow \text{NNS, 0.75}), \\
&\quad (\text{NP-OBJ} \rightarrow \text{DET NN, 0}), \\
&\quad (\text{NP-OBJ} \rightarrow \text{DET JJ NNS, 0.25})
\end{aligned}$$

Definition 3 PCFG Partitioning

Given a PCFG $G = (W, N, N_S, R)$ and a partition Π_N of the set of nonterminals N , the PCFG derived by partitioning G with Π_N is $G' = (W, \Pi_N, N_S, R')$, where:

$$R' = \left\{ (a_1 \rightarrow a_2 \dots a_n, p) \mid \begin{array}{l} \{(b_1^1 \rightarrow b_2^1 \dots b_n^1, p^1), \dots (b_1^m \rightarrow b_2^m \dots b_n^m, p^m)\} \in \Omega, \\ a_1 \in \Pi^N, b_i^j \in a_i, \\ \forall i, 2 \leq i \leq n \text{ (either } a_i = b_i^j \in W, \text{ or } a_i \in \Pi^N, b_i^j \in a_i), \\ p = (\sum_{j=1}^m p^j) / |a_1| \end{array} \right\}, \text{ and}$$

Ω is a partition of R such that each $O \in \Omega$ contains all and only elements from R $(b_1^1 \rightarrow b_2^1 \dots b_n^1, p^1), \dots (b_1^m \rightarrow b_2^m \dots b_n^m, p^m)$ for which the following holds:

$$\forall i, 1 \leq i \leq n \text{ (either } b_i^1 = b_i^2 = \dots = b_i^m \in W, \text{ or } \{b_i^1, b_i^2, \dots, b_i^m\} \subseteq P, P \in \Pi_N).$$

With rules of zero probability removed, G' is now identical to the original grammar G in the example in the previous section.

2.3. Partition Search

A PCFG together with merge and split operations on the nonterminal set defines a space of derived grammars which can be searched for a new PCFG that optimises some given objective function. The disadvantage of this search space is that it is infinite, and each split operation requires the reestimation of rule probabilities from a training corpus, making it computationally much more expensive than a merge operation.

However, there is a simple way to make the search space finite, and at the same time to make split operations redundant. The resulting method, Grammar Learning by Partition Search, is described in this section. First, the merge operation that was informally introduced in the last section is generalised and defined formally. Next, search space, search task and objective function are discussed, and finally, the search algorithm is presented.

2.3.1. PCFG Partitioning

An arbitrary number of merges can be represented by a partition of the set of nonterminals. For the example presented in Section 2.2.1. above, the partition of the nonterminal set N in G that corresponds to the nonterminal set N' in G' is $\{S\}, \{NP-SBJ, NP-OBJ\}, \{VP\}$. The original grammar G together with a partition of its nonterminal set fully specifies the new grammar G' : the new rules and probabilities, and the entire new grammar G' can be derived from the partition together with the original grammar G . The process of obtaining a new grammar G' , given a base grammar G and a partition of the nonterminal set N of G will be called PCFG Partitioning⁴.

⁴The concept of context-free grammar partitioning in this paper is not directly related to that in (Korenjak, 1969; Weng and Stolcke, 1995), and later publications by Weng et al. In these previous approaches, a non-probabilistic CFG's *set of rules* is partitioned into subsets of rules. The partition is drawn along a specific nonterminal NT , which serves as an interface through which the subsets of rules (hence, subgrammars) can communicate after partition (one grammar calling the other). In the calling subgrammar, NT in RHSS is prefixed *vt* to denote that it is a 'virtual terminal'. In the following example from (Luk et al., 2000, p. +2), partitioning grammar G along the nonterminal NP yields subgrammars

In the examples in Sections 2.2.1. and 2.2.2., a notational convention was tacitly adopted which is also used in the formal definition of PCFG Partitioning (Definition 3). As a result of merging, NP-SUBJ and NP-OBJ become a single new nonterminal. This new nonterminal was represented above by the set of merged nonterminals $\{NP-SBJ, NP-OBJ\}$ (in the partition), as well as by a new symbol string NP (in the definition of G'). The two representations are treated as interchangeable: the new nonterminal is represented either as a set or a nonterminal symbol⁵.

The definition of PCFG Partitioning can be paraphrased as follows⁶. Given a PCFG $G \mathbb{W}, (N, N_S, R)$ and a partition Π_N of the set of nonterminals N , the PCFG derived from G by Π_N is $G' = (W, \Pi_N, N_S, R')$. That is, the set of terminals remains the same, and the new set of nonterminals is just the partition Π_N ⁷. Ω is the partition of R in which all production rules are grouped together that become identical as a result of the nonterminal merges specified by Π_N . Then, the new set of probabilistic rules R' contains one element $(a_1 \rightarrow a_2 \dots a_n, p)$ for each element $\{(b_1^1 \rightarrow b_2^1 \dots b_n^1, p^1), \dots (b_1^m \rightarrow b_2^m \dots b_n^m, p^m)\}$ of Ω , such that the following holds between them: a_1 is a nonterminal from Π^N and contains all the b_1^j ; for the other a_i , either a_i is a nonterminal and contains all the b_i^j , or it is a terminal and is identical to b_i^j . The new rule probability p is the sum of all probabilities $p^j, 1 \leq j \leq m$ renormalised by the size of the set a_1 .

2.3.2. Search space

As stated previously, the search space for Grammar Learning by Partition Search can be made finite and search-

G_S and G_{NP} as follows:

$G :$	$G_S :$	$G_{NP} :$
1. $S \rightarrow NP VP$	INPUT = {vtNP}	INPUT = \emptyset
2. $NP \rightarrow n$	OUTPUT = {S}	OUTPUT = {NP}
3. $NP \rightarrow det n$	1. $S \rightarrow vtNP VP$	1. $NP \rightarrow n$
4. $NP \rightarrow NP PP$	2. $VP \rightarrow v vtNP$	2. $NP \rightarrow det n$
5. $PP \rightarrow prep NP$	3. $NP \rightarrow prep vtNP$	3. $NP \rightarrow NP PP$
6. $VP \rightarrow v NP$		

⁵The name assigned to the new nonterminal in the current implementation of PCFG Partitioning is the longest common prefix of the old nonterminals that are merged followed by an indexation tag (to distinguish otherwise identical names).

⁶This definition is for PCFGs with one start symbol. In the current implementation of Partition Search, PCFGs are permitted to have multiple start symbols and these can be merged with other nonterminals. The probability of a new start symbol resulting from a merge is the renormalised sum of the probabilities of only the start symbols participating in the merge.

⁷Recall previous comments about this notational convention.

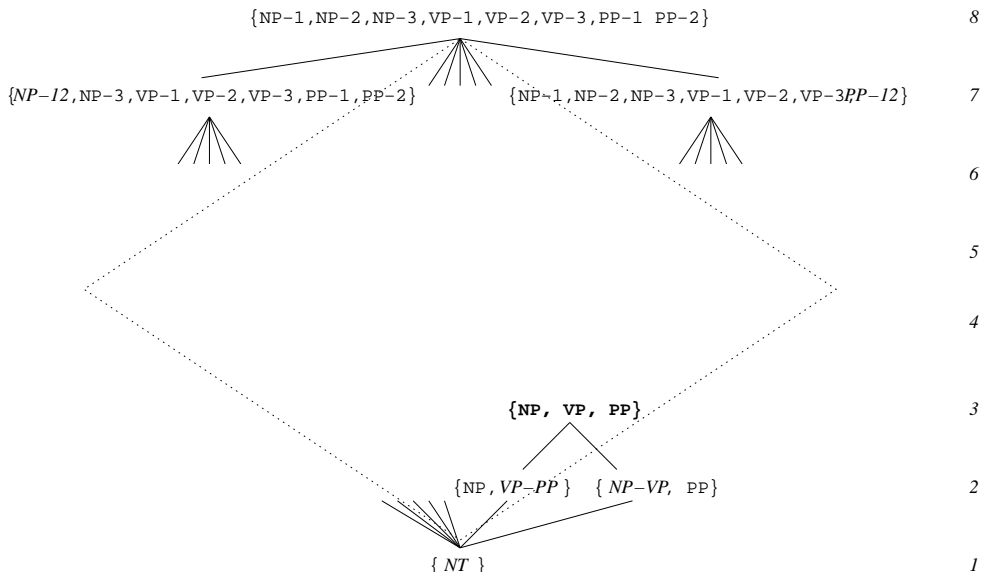


Figure 1: Simple example of a partition search space.

able entirely by merge operations (grammar partitions).

Making the search space finite: The number of merge operations that can be applied to a nonterminal set is finite, because after some finite number of merges there remains only one nonterminal. On the other hand, the number of split operations that can sensibly be applied to a nonterminal NT has a natural upper bound in the number of different terminal strings dominated by NT in a corpus of evidence (e.g. the corpus the PCFG was trained on). For example, when splitting the nonterminal NP into subjects and objects, there would be no point in creating more new nonterminals than the number of different subjects and objects found in the corpus.

Given these bounds, there is a finite number of distinct grammars derivable from the original grammar by different combinations of merge and split operations. This forms the basic space of candidate solutions for Grammar Learning by Partition Search.

Making the search space searchable by grammar partitioning only: Imposing an upper limit on the number and kind of split operations permitted not only makes the search space finite but also makes it possible to directly derive the *maximally split nonterminal set* (Max Set). Once the Max Set has been defined, the single grammar corresponding to it — the *maximally split Grammar* (Max Grammar) — can be derived and retrained on the training corpus⁸.

The set of points in the search space corresponds to the set of partitions of the Max Set. Search for an optimal grammar can thus be carried out directly in the partition space of the Max Grammar.

Structuring the search space: The finite search space can be given hierarchical structure as shown in Figure 1

⁸This can be done as follows: for each nonterminal N , count the number n of different terminal strings it dominates in the training corpus, and tag each occurrence of NT with a number $NT-1, \dots, NT-n$; duplicate the rules containing NT correspondingly, and calculate the rule probabilities.

for an example of a very simple base nonterminal set $\{NP, VP, PP\}$, and a corpus which contains three different NPs, three different VPs and two different PPs.

At the top of the graph is the Max Set. The sets at the next level down (level 7) are created by merging pairs of nonterminals in the Max Set, and so on for subsequent levels. At the bottom is the *maximally merged nonterminal set* (Min Set) consisting of a single nonterminal NT . The sets at the level immediately above it can be created by splitting NT in different ways. The sets at level 2 are created from those at level 1 by splitting one of their elements. The original nonterminal set ends up somewhere in between the top and bottom (at level 3 in this example).

While this search space definition results in a finite search space and obviates the need for the expensive split operation, the space will still be vast for all but trivial corpora. In Section 3.3. below, alternative ways for defining the Max Set are described that result in much smaller search spaces.

2.3.3. Search task and evaluation function

The input to the Partition Search procedure consists of a base grammar G_0 , a base training corpus C , and a task-specific training corpus D^T . G_0 and C are used to create the Max Grammar G . The **search task** can then be defined as follows:

Given the maximally split PCFG $G = (W, N, N_S, R)$, a data set of sentences D , and a set of target parses D^T for D , find a partition Π_N of N that derives a grammar $G' = (W, \Pi_N, N_S, R')$, such that $|R'|$ is minimised, and $f(G', DD^T)$ is maximised, where f scores the performance of G' on D as compared to D^T .

The size of the nonterminal set and hence of the grammar decreases from the top to the bottom of the search space. Therefore, if the partition space is searched top-down, grammar size is minimised automatically and does not need to be assessed explicitly.

In the current implementation, the **evaluation function** f simply calculates the F-Score achieved by a candidate

grammar on D as compared to D^T . The F-Score is obtained by combining the standard PARSEVAL evaluation metrics *Precision* and *Recall*⁹ as follows: $2 \times \textit{Precision} \times \textit{Recall} / (\textit{Precision} + \textit{Recall})$.

An existing parser¹⁰ was used to obtain Viterbi parses. If the parser failed to find a complete parse for a sentence, a simple grammar extension method was used to obtain partial parses instead (Schmid and Schulte im Walde (2000, p. 728) use an almost identical method).

2.3.4. Search algorithm

Since each point in the search space can be accessed directly by applying the corresponding nonterminal set partition to the Max Grammar, the search space can be searched in any direction by any search method using partitions to represent candidate grammars.

In the current implementation (see pseudo-code representation in Procedure 1), a variant of beam search is used to search the partition space top down. A list of the n current best candidate partitions is maintained (initialised to the Max Set). For each of the n current best partitions a random subset of size b of its children in the hierarchy is generated and evaluated. From the union of current best partitions and the newly generated candidate partitions, the n best elements are selected and form the new current best set. This process is iterated until either no new partitions can be generated that are better than their parents, or the lowest level of the partition tree is reached. In each iteration the size of the nonterminal set decreases by one.

The size of the search space grows exponentially with the size i of the Max Set. However, the complexity of the Partition Search algorithm is only $O(nbi)$, because only up to $n \times b$ partitions are evaluated in each of up to i iterations¹¹.

3. Grammar Optimisation with Partition Search

3.1. Testing and Training Data

Sections 15–18 of WSJC were used for deriving the base grammar and as the base training corpus, and different randomly selected subsets of Section 1 from the same corpus were used as task-specific training corpora during search. Section 20 was used for final performance tests.

The Brill Tagger was used for POS tagging testing data, and achieved an average accuracy of 97.5% (as evaluated by evalb).

3.2. Base grammar

A simple treebank grammar¹² was derived from Sections 15–18 of the WSJ corpus by the following procedure:

⁹I used the evalb program by Sekine and Collins (<http://cs.nyu.edu/cs/projects/proteus/evalb/>) to obtain Precision and Recall figures.

¹⁰LoPar (Schmid, 2000) in its non-head-lexicalised mode. Available from <http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/LoPar-en.html>.

¹¹As before, n is the number of current best candidate solutions, b is the width of the beam, and i is the size of the Max Set.

¹²The term was coined by Charniak (1996).

Procedure 1

P-SEARCH($(W, N, N_S, R), D, D^T, n, x, b$)

```

1: Stop ← FALSE
2:  $P_{intermediate} \leftarrow \text{INITIALISE}(N)$ 
3: EVALUATE( $P_{intermediate}, G, D, D^T, x$ )
4: while not Stop do
5:    $P_{current} \leftarrow \text{SELECT}(P_{intermediate})$ 
6:    $P_{new} \leftarrow \text{GENERATE}(P_{current}, b)$ 
7:   if  $P_{new} = \text{EMPTYLIST}$  then
8:     Stop ← TRUE
9:   else
10:     $P_{intermediate} \leftarrow \text{APPEND}(P_{current}, P_{new})$ 
11:    EVALUATE( $P_{intermediate}, G, D, D^T, x$ )
12:   end if
13: end while
14: return  $P_{current}$ 
15:
16: Subprocedure INITIALISE( $N$ )
17: return set containing trivial partition of  $N$ 
18:
19: Subprocedure SELECT( $P_{intermediate}, n$ )
20: return  $n$  best elements from  $P_{intermediate}$ 
21:
22: Subprocedure GENERATE( $P_{current}, b$ )
23: Rtn rnVal ← EMPTYLIST
24: for all  $p \in P_{current}$  do
25:   List ← generate  $b$  random elements of
     {  $\Pi \mid \Pi$  is a partition of  $p$  and  $|\Pi| = |p| - 1$  }
26:   Rtn rnVal ← APPEND(Rtn rnVal, List)
27: end for
28: return Rtn rnVal
29:
30: Subprocedure EVALUATE( $P_{intermediate}, G, D, D^T, x$ )
31: for all  $p \in P_{intermediate}$  do
32:    $G' \leftarrow$  partition grammar  $G$  with  $p$ 
33:    $D^A \leftarrow$  parse data  $D$  with  $G'$ 
34:   score of  $p$  is F-Score of  $D^A$  against  $D^T$ 
35: end for

```

- Iteratively edit the corpus by deleting (i) brackets and labels that correspond to empty category expansions; (ii) brackets and labels containing a single constituent that is not labelled with a POS-tag; (iii) cross-indexation tags; (iv) brackets that become empty through a deletion.
- Convert each remaining bracketing in the corpus into the corresponding production rule.
- Collect sets of terminals W , nonterminals N and start symbols N_S from the corpus. Probabilities p for rules $n \rightarrow \alpha$ are calculated from the rule frequencies C by Maximum Likelihood Estimation: $p(n \rightarrow \alpha) = \frac{C(n \rightarrow \alpha)}{\sum_i C(n \rightarrow \alpha^i)}$.

This procedure creates the base grammar *BARE* which has 10,118 rules and 147 nonterminals.

3.3. Restricting the search space further

The simple method described in Section 2.3.2. for defining the maximally split nonterminal set (Max Set) tends to result in vast search spaces. Using parent node (PN) information to create the Max Set is much more restrictive and linguistically motivated. The Max Grammar *PN* used in the experiments reported below can be seen as making use of *Local Structural Context* (Belz, 2001): the independence assumptions inherent in PCFGs are weakened by making

the rules’ expansion probabilities dependent on part of their immediate structural context (here, its parent node). To obtain the grammar *PN*, the base grammar’s nonterminal set is maximally split on the basis of the *parent node* under which rules are found in the base training corpus¹³. Several previous investigations have demonstrated improvement in parsing results due to the inclusion of parent node information (Charniak and Carroll, 1994; Johnson, 1998b; Verdú-Mas et al., 2000).

Another possibility is to use the base grammar *BARE* itself as the Max Grammar. This is a very restrictive search space definition and amounts to an attempt to optimise the base grammar in terms of its size and its performance on a given task without adding any information. Results are given below for both *BARE* and *PN* as Max Grammars.

In the current implementation of the algorithm, the search space is reduced further by avoiding duplicate partitions, and by only allowing merges of nonterminals that have the same phrase prefix $NP-*$, $VP-*$ etc.

The Max Grammars end up having sets of nonterminals that differ from the bracket labels used in the WSJC: while the phrase categories (e.g. *NP*) are the same, the tags (e.g. $*-S$, $*-3$) on the phrase category labels may differ. In the evaluation, all labels starting with the same phrase category prefix are considered equivalent.

3.4. Optimising the WSJ treebank grammar

Base grammar *BARE* achieves an F-Score of 74.09 on the full parsing task. Since the base grammar is just the treebank grammar extracted from the WSJC, and the task it is optimised for is full parsing, the most directly related research is other treebank grammar research. Over the last few years, a range of research projects — e.g. Charniak (1996), Cardie & Pierce (1998), Johnson (1998a, 2000), Krotov et al. (2000) — have looked at probabilistic grammars that have been directly derived from bracketed corpora.

Because the number of rules in treebank grammars is very large at least in the case of the WSJC, and because their parsing performance moreover tends to be not very good, some techniques are usually applied to reduce grammar size and to improve performance. All approaches edit the corpus in some way, e.g. eliminating single child rules, empty category rules, functional tags, co-indexation tags, and punctuation marks. Different compression methods (such as eliminating rules with frequency less than some n) have been investigated that reduce the size of grammars without too much loss of performance (in particular by Charniak and Krotov et al.). To improve parsing performance, e.g. Charniak relabels auxiliary verbs with a separate POS-tag and incorporates a “right-branching correction” into the parser to make it prefer right-branching structures.

Several other grammar building and training methods are similar to treebank grammar construction: Bod & Scha’s Data-Oriented Parsing method which extracts tree fragments rather than rules from corpora, and Memory-

¹³The parent node of a phrase is the category of the phrase that immediately contains it.

Based Learning methods (Daelemans et al.) for building parsing systems from corpora.

Table 1 shows results achieved by Partition Search with grammars *BARE* and *PN* as Max Grammars. The first column shows the Max Grammar used in a given batch of experiments. The second column indicates the type of result, where the Max Grammar result is the F-Score, grammar size and number of nonterminals of the Max Grammar itself, and the remaining results are the average and single best results achieved by Partition Search. The third and fourth columns show the number of iterations and evaluations carried out before search stopped. Columns 5–8 show details of the final solution grammars: column 5 shows the evaluation score on the training data, column 6 the overall F-Score on the testing data, column 7 the size, and the last column gives the number of nonterminals.

The best size reduction result was 35 nonterminals and 8,409 rules with an increase in the F-Score to 74.54 (as compared to the baseline of 74.09). The best performance increase result was an F-Score of 78.01 (compared to the baseline of 74.09), accompanied by an increase in the number of rules to 15,608.

The best performance result reported here is better than the best results reported by Charniak (1996) and Krotov et al. (2000), even though the previous results were obtained after using ca. 10/11 of the WSJ corpus as a training set compared to 3/25 used here (UF = unlabelled F-Score, LF = labelled F-Score):

	UF	LF
Krotov et al. (2000)	79.12	76.09
Charniak (1996)	79.59	–
Optimised <i>PN</i> -Grammar	80.74	78.01

During Partition Search, only those nonterminal subsets are merged that result in an improved or unchanged F-Score. In the case of the grammar *BARE* as the Max Grammar, merging means eliminating phrase subcategories reflecting grammatical function. As can be seen from the results in Table 1, a lot of grammatical function information can be eliminated without affecting parsing results: of the 147 original phrase subcategories, only just over a third remain on average. In the case of the *PN* Max Grammar, search found it a lot harder to eliminate parent node subcategories without worsening parsing performance: just over half of the 970 parent node subcategories remain on average. The biggest part of the improvement in parsing performance is due to way the Max Grammar is defined, i.e. to the addition of parent node information (from 74.09 to 77.8 F-Score).

3.5. General comments

Partition Search is able to reduce grammar size by merging groups of nonterminals (hence groups of rules) that do not need to be distinguished for a given task. It is able to improve parsing performance firstly by grammar generalisation (a partitioned grammar parses a superset of the sentences parsed by the base grammar), and secondly by reranking parse probabilities (the most likely parse for a sentence under a partitioned grammar can differ from its most likely parse under the base grammar).

Max Grammar		Iter.	Eval.	F-Score (subset)	F-Score (WSJC S 1)	Size (rules)	Nonterms
<i>BARE</i>	Max Grammar result:				74.09	10,118	147
	Average:	90.2	2,000.8	83.43	74.05	9,288.4	59.8
	Best (size and F-score):	114	2,686	86.04	74.54	8,409	35
<i>PN</i>	Max Grammar result:				77.8	16,480	970
	Average:	426.8	10,559.4	87.8734	77.87	15,850	545.2
	Best (size):	656	16,335	91.38	77.81	15,403	316
	Best (F-score):	625	15,554	89.44	78.01	15,608	347

Table 1: Results for partition tree search on full parsing task, WSJC Section 1 (averaged over 5 runs, variable parameters: $x = 100$, $b = 5$, $n = 5$).

Automatic methods for optimising and adapting grammars for new tasks and domains are particularly useful because context-free grammars cannot be learnt from scratch from data. At the very least, an upper bound must be placed on the number of nonterminals allowed. Even when that is done, there is no likelihood that the grammars resulting from an otherwise unsupervised method will look anything like a linguistic grammar whose parses can provide a basis for semantic analysis. The present method preserves the basic phrase structures and categories of the base grammar, while reranking its parse sets and eliminating phrase distinctions not needed for a given task.

Preliminary tests revealed that results were surprisingly constant over different combinations of variable parameter values, although training subset size of less than 50 meant unpredictable results for the complete WSJC Section 1. For a random subset of size 50 and above, there is an almost complete correspondence between subset F-Score and Section 1 F-Score, i.e. higher subset F-Score almost always means higher Section 1 F-Score.

Unlike other grammar compression methods (Charniak, 1996; Krotov et al., 2000), Partition Search achieves loss-less compression, in the sense that the compressed grammars are guaranteed to be able to parse all of the sentences parsed by the original grammar.

Compared to other approaches using parent node information (Charniak and Carroll, 1994; Johnson, 1998b; Verdú-Mas et al., 2000), the approach presented here has the advantage of being able to select a subset of all parent node information on the basis of its usefulness for a given parsing task. This saves on grammar complexity, hence parsing cost.

4. Conclusions and Further Research

Grammar Learning by Partition Search was shown to be an efficient method for constructing PCFGs optimised for a given parsing task. The best grammar constructed by Partition Search in the reported experiments outperforms the best existing results for nonlexicalised parsing by a significant margin. In the same experiments, grammar size was reduced by up to 16.89%. Partition Search has the advantage of reducing grammar size without loss of grammar coverage while achieving an improvement in grammar performance.

In future research, the P_SEARCH procedure will be used as a testbed for comparing the performance of differ-

ent search techniques, including greedy depth-first search and genetic search (only some of the subprocedures need to be replaced for each new search type). Further research will also look at additionally incorporating lexicalisation, and testing a wider set of variable parameter combinations.

5. Acknowledgements

The research reported in this paper was in part funded under the European Union’s TMR programme (Grant No. ERBFMRXCT980237).

6. References

- A. Belz. 2001. Optimising corpus-derived probabilistic grammars. In *Proceedings of Corpus Linguistics 2001*, pages 46–57.
- A. Belz. 2002. Learning grammars for noun phrase extraction by partition search. In *Proceedings of LREC Workshop on Event Modelling for Multilingual Document Linking*.
- Claire Cardie and Darren Pierce. 1998. Error-driven pruning of treebank grammars for base noun phrase identification. In *Proceedings of COLING-ACL ’98*, pages 218–224.
- Eugene Charniak and Glenn Carroll. 1994. Context-sensitive statistics for improved grammatical language models. Technical Report CS-94-07, Department of Computer Science, Brown University.
- Eugene Charniak. 1996. Tree-bank grammars. Technical Report CS-96-02, Department of Computer Science, Brown University.
- M. Johnson. 1998a. The effect of alternative tree representations on tree bank grammars. In *Proceedings of the Joint Conference on New methods in Language Processing and Computational Natural Language Learning (NeMLaP-3/CoNLL’98)*, pages 39–48.
- Mark Johnson. 1998b. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.
- A. J. Korenjak. 1969. A practical method for constructing LR(k) processors. *Communications of the ACM*, 12(11).
- A. Krotov, M. Hepple, R. Gaizauskas, and Y. Wilks. 2000. Evaluating two methods for treebank grammar compaction. *Natural Language Engineering*, 5(4):377–394.
- Po Chui Luk, Helen Meng, and Fuliang Weng. 2000. Grammar partitioning and parser composition for natural language understanding. In *Proceedings of ICSLP 2000*.

- H. Schmid and S. Schulte Im Walde. 2000. Robust German noun chunking with a probabilistic context-free grammar. In *Proceedings of COLING 2000*, pages 726–732.
- H. Schmid. 2000. LoPar: Design and implementation. Bericht des Sonderforschungsbereiches “Sprachtheoretische Grundlagen für die Computerlinguistik” 149, Institute for Computational Linguistics, University of Stuttgart.
- Jose Luis Verdú-Mas, Jorge Calera-Rubio, and Rafael C. Carrasco. 2000. A comparison of PCFG models. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 123–125.
- F. L. Weng and A. Stolcke. 1995. Partitioning grammars and composing parsers. In *Proceedings of the 4th International Workshop on Parsing Technologies*.

A Semantic-driven Approach to Hypertextual Authoring

R. Basili, A. Moschitti, M.T. Pazienza, F.M. Zanzotto

University of Rome Tor Vergata,
Department of Computer Science, Systems and Production,
00133 Roma (Italy),
{basili, moschitti, pazienza, zanzotto}@info.uniroma2.it

Abstract

Even if the use of the hypertext paradigm is nowadays very diffused, its potential benefits are not completely exploited by the community of the users. This is particularly evident in the case of the news agencies. The major reasons for the above limitation are the high costs for manually creating and maintaining the sets of complete links of a large-scale hypertext. This is especially true for news agencies. Therefore, in this paper we propose a method to address the problem of the automatic construction of the hyper-links based on Information Extraction techniques that enable documents (mainly news items) to be represented in a canonical form, hereafter called *objective representation* (OR). Our hyper-linking method is presented after the analysis of the traditional approaches to the same problem. We will describe the notion of objective representation and the formalism to express the linking constraints. Finally, we will sketch our future research work in the area.

1. Introduction

Even if the use of the hypertext paradigm is nowadays very diffused, its potential benefits are not completely exploited by the community of the users. This is particularly evident in the case of the news agencies. A survey, reported in (Outing, 1996), found that there were 1,115 commercial newspaper online services world-wide, 94% of which used a simplified version of hypertext which does not provide the full use of the hypertext capabilities of the WWW. The users may be able to navigate to a particular article in the current edition of an online paper by using hypertext links, but they must then read the entire article to find the information that interests them. The documents are dead ends in the hypertext, rather than offering starting points for explorations. In order to truly reflect the hypertext nature of the Web, links should to be placed within and between the documents.

The major reasons for the above limitation is, as (Westland, 1991) has pointed out, the high costs for manually creating and maintaining the sets of complete links of a large-scale hypertext. This is especially true for news agencies, given the volume of articles produced every day. Aside from the time-and-money aspects of building such large hypertexts manually, humans are inconsistent in assigning hypertext links between the paragraphs of documents (Ellis et al., April 1994; Green, 1997). That is, different linkers disagree with each other as to where to insert hypertext links into a document.

The cost and inconsistency of manually constructed hypertexts does not necessarily mean that large-scale hypertexts can never be built. It is well known in the IR community that humans are inconsistent in assigning index terms to documents, but this has not hindered the construction of automatic indexing systems intended to be used for very large collections of documents.

The taxonomy of link types given in (Allan, 1995) is very useful to understand the problem of the automatic construction of hyperlinks since it classifies links according to the abilities required for an eventual manual construction. Links are classified according the following

classes:

- *Pattern Matching links*, which are easy link to discovered as they can be found through a pattern-matching algorithm. An example of these is glossary links or links between proposition.
- *Automatic links*, which can be in part captured by traditional Information Retrieval techniques. For example links among documents discussing about the same topics.
- *Manual links*, which require text analysis at level of Natural Language Understanding.

While the first two types of links have been approached successfully the third one is judged by Allan (Allan, 1995) to be inaccessible to automatic hypertext construction.

In this paper we propose a method to address the problem of the automatic construction of the "manual" links as defined in (Allan, 1995). The proposed method is based on Information Extraction techniques that enable documents (mainly news items) to be represented in a canonical form, hereafter called *objective representation* (OR). This latter describes some of the important information contained in the documents, mainly the named entities and the domain events found in the target document. Therefore, this document representation allows to draw more motivated inter-document hyper-links since a declarative language for describing linking constraints can be settled over it. Linking rules, i.e. the rules that justify a link between two documents, are in fact written as constraints over the related ORs. The detection of the domain events and of the named entities relies on a knowledge-based IE system composed by a robust parser (Basili et al., 2000b) and a discourse interpreter (Gaizauskas and Humphreys, 1997). As any IE system, this linking methodology requires a large domain knowledge base. The overall approach foresees the methods for the automatic extraction of this knowledge in an unsupervised fashion (Basili et al., 2000a; Basili et al., 2002).

Our hyper-linking method is presented in Sec. 3. after the analysis of the traditional approaches to the same problem (Sec. 2.). We will describe the notion of objective representation and the formalism to express the linking constraints. Finally, we will sketch the future work (Sec. 4.).

2. Traditional Approaches

In literature the automatic construction of hypertext is based on classical *IR* techniques to measure the relatedness of document couples. Only a *bag of words* are used for expressing the document contents. This results in a poor set of link type manageable in automatic way. In (Allan, 1995) is presented a reformulated taxonomy of links (Trigg, 1983) in order to identify the link type achievable with an automatic approaches. The set of link type has been divided into three major categories based upon whether or not their identification can be carried out automatically (with the *IR* current technology). The three categories are *Pattern-matching*, *Automatic* and *Manual*. Unfortunately, some types of links straddle the boundaries of the taxonomy, depending upon the document collection being linked.

Pattern-matching Links is a large class of link types. They can be found easily using simple pattern-matching techniques. An obvious example of such a link type is *definition* that can be found by matching words in a document to entries in a dictionary. In almost cases, these links are from a word or phrase to a small documents. They do not take into account the context of the definition so the destination document may be the same for the word or phrase searched for; no matter where the word or phrase occurs. Structural links belong to the pattern-matching category. They are those that represent layout or possibly logical structure of a document. For example, links between chapters or sections, links from a reference to a figure to the figure itself, and links from a bibliographic citation to the cited work, are all structural links. They can be discovered by mark-up codes embedded in the text. Pattern-matching links form a class that is computationally simple for automatic detection.

Automatic Links are links which cannot typically be located trivially using patterns, but which the automatic *IR* techniques can identify with marked success. Typical automatic links that can be identified are:

- *Revision links* are a fairly straightforward class of relationship between texts, including both ancestor and descendent relationships.
- *Summary and expansion links* are inverses of one another. A summary link type is attached to a link that starts at a discussion of a topic and has as its destination a more condensed discussion of the same topic. Equivalence links represent strongly related discussions of the same topic.
- *Tangent links* are equivalence links that relate topics in an unusual or tangential manner (often by comparison with other links). For example, a link from a document about *Sivlio Berlusconi* as Italy Prime Minister to one about Milan football club (whose Berlusconi is the president) would be a tangential link.

- *Aggregate links* are those that group together several related documents. An aggregate link may in fact have several destinations, allowing the destination documents to be treated as a whole when desirable.

Manual links are those which are judged by the *IR* community unable to be located without human intervention. The natural language understanding researchers have had some significant success within constrained subject areas, so some manual links could be automatically described within those limited domains. Unfortunately, those techniques are not yet extensible to a general setting, so this class of link types seems to remain inaccessible to automatic approaches. Manual links include those which connect documents which describe circumstances under which one document occurred, those which collect the various components of a debate or argument, and those that describe forms of logical implication (caused-by, purpose, warning, and so on).

2.1. A more semantic based approach

An attempt to extend the boundaries of automatic links towards the manual links has been done in (Green, 1997). In this work an automatic method for the construction of hypertext links via *lexical chains* has been carried out. Lexical chains capture the semantic relations between words that occur throughout a text. Each chain is a set of related words that captures a portion of the cohesive structure of a text. By considering the distribution of chains within an article it is possible to build links between the paragraphs. A link is activated if the similarity score of the chains contained in two different articles overcome a predefined threshold. The method comprises three steps: determining the lexical chains in a text, building links between the paragraphs of articles, and building links between articles. A comparison of this methodology with the traditional *IR* techniques resulted in higher user satisfaction. Lexical chains allow to retrieve a wider set of link type. As an example let us consider two documents that speak about the same fact with different words. The scalar product (a wide used *IR* metrics in the Vector Space Model) between the two documents would be very low as the documents have different *bag of words*. This prevents the activation of a relatedness link. On the contrary lexical chains refer to the meaning of words. They use synonyms of words in texts so their similarity between documents will be higher.

Lexical chains seems to solve some of *IR* problem in discovering links but some problems remain unsolved:

- The link type of two documents, which have similar lexical chains, is unknown. We could claim as an explanation that the documents contain some related semantic information. However this explanation is too generic as it is valid for each generated link.
- *Consequence links* remain unsolved. It is not possible specify the consequence relation between two documents for two main reasons: a) The lexical chains of the premised tend to be very different from the consequence. b) These links are directional while the similarity between chains is symmetric.

- Ambiguity and data sparseness affect the precision in discovering valid chains. So we can expect a lot of wrong links.

In next Section it is presented a different approach that solve the two first problems. It provides a methodology for capturing the unsolved link as well as the explanation for them. The third problem has been bound using domain knowledge for conceptualise the information.

3. A "semantic-driven" hyper-linking method

The above approaches mainly relate documents if they are enough similar according to the chosen document representation space, i.e. the bag-of-word abstraction or the lexical chain model. Therefore, according to these approaches "relatedness" is the only reason why two documents may be hyper-linked together. However, this notion of relatedness does not give the possibility of defining user-oriented hypertexts. Each user has to be exposed to the same hypertext regardless his information needs. For instance, the above approaches may relate the two news items in Fig. 1 because of the fact that in the two documents the *Intel* stem increases the relatedness of the two documents. However, the link user is not aware of the reason why the two documents are related and, while reading the first news item, he has not hints that may suggest if the related news article is of any interest to him. The justification of the link may be more easily highlighted if the domain relevant information is captured, i.e. the fact that both the first item and the second one describe an *Intel acquisition activity*.

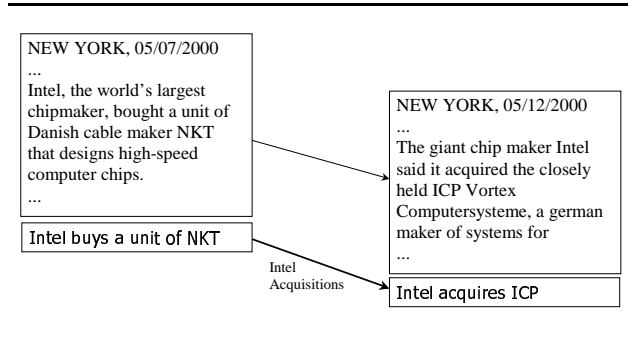


Figure 1: An example of justified link

The facts justifying the hyper-link between the two documents are respectively:

- Intel buys a unit of NKT
- Intel acquires ICP

It is worth noticing that a very precise information is needed for linking the two documents, i.e. the "equivalence" of *buy* and *acquire*. This information may be also used in an IR based hyper-linker using a query expansion technique but the justification of the link is still very difficult.

Furthermore, this notion of relatedness limits the possibility of linking documents. For instance in Fig. 2,

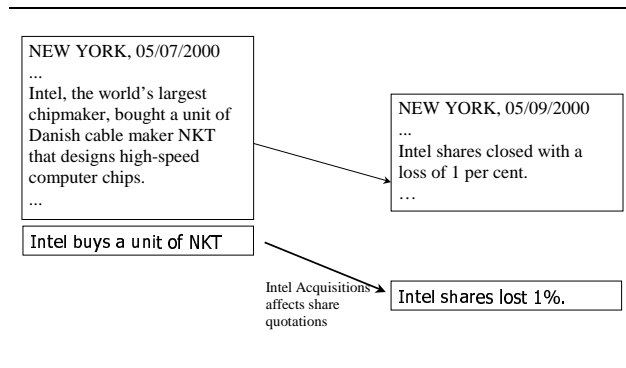


Figure 2: A complex justified link

the link between the two documents is justified by the fact that an *Intel acquisition* affects the *share prices* of a particular period of time. The facts justifying such a kind of document relation are respectively:

- Intel buys a unit of NKT
- Intel shares lost 1%

Such a kind of link is very difficult to capture if the analysis is not based on the more structured document representation.

The automatic hyper-linking method we propose is then based on an abstraction of the document, the objective representation (OR) that describes in a canonical form the salient information carried by the document. This objective representation, due to its nature, may be also considered language independent. Therefore, it enables the automatic hyper-linking between documents of different languages. Both his canonical representation, i.e. the OR, and the language for defining the linking constraints are described in the following sections.

3.1. The objective representation

The quality of the hyper-links that may be drawn in such a method strictly depends on the assumed representation of the document content. Furthermore, it is crucial that the intended information is actually captured by the IE system.

The objective representation we have defined is not too far from the actual document content and aims to represent the relevant document information with respect to a given knowledge domain. In particular, given a document D , its OR contains the named entities and the main events of the document D . These latter mainly represent particular domain relevant verb phrases that appear in the document. Both the named entities and the events are classified according to a knowledge representation scheme related to a target domain.

The objective representation is then a couple $OR(D) = (NEs, Events)$ where NEs is the set of the categorised named entities of D while the $Events$ is the set of the categorised events. Each event in $Events$ has the following form:

$$EventType(Verb, Arguments) \quad (1)$$

where *EventType* is the type of the event, *Verb* is the actual verb that appears in the document and *Args* are the arguments of the verb according to the event type. Each argument representation carries its syntactic/semantic relation, the actual lexical of its semantic governor, and the type of this latter. For instance, the documents in Fig. 1 should contain respectively in their ORs the following events:

- `buy_event(agent(company, Intel), patient(object, a_unit_of_NKT))`
- `buy_event(agent(company, Intel), patient(company, ICP))`

Naturally, the efficacy of the OR strictly depends on the nature of the information that is contained in the knowledge base. The method for extracting such a knowledge and for the definition of the equivalence between different surface forms is described in (Basili et al., 2002).

3.2. Typing links using events: a declarative formalism

Once an objective representations of documents are available it is possible to write down a set of rules that can activate several links that traditional IR techniques (see Section 2.) cannot capture. However it is not possible to define general linking rules valid for each domains and for each user needs. As an example consider two documents: d_0 that speaks about Ferrari race in the grand prix of Imola and d_1 in which it is stated that FIAT market shares increase their quotation. If a user wants know all the facts which cause the event in d_1 (e.g. the document d_0) some knowledge about the correlation between FIAT and Ferrari have to be drawn (i.e. Ferrari is a part of FIAT and winning a race increases the share value of a Company).

Thus a systems that really wants to afford hypertext construction including links of third type (see Section 2.) should provide both a set of general rules and a set of specific rules. Moreover, the specific rules should be customisable to satisfy a wide range of user needs. These rules will be then used by the linking algorithm to draw links among documents.

3.2.1. The linking rule formalism

We have adopted a declarative formalism in which the rules and the knowledge required are easy to be written by the final user. The rules are expressed by a logical formalism.

The events in the *OR* are coded by means of Prolog predicates of the following type:

```
ev(EVENT_CATEGORY, EVENT_LEX, [
  arg(AGENT, AGENT_CATEGORY,
      AGENT_LEX),
  arg(DIROBJ, DIROBJ_CATEGORY,
      DIROBJ_LEX),
  arg(MODIFIER1, HANDLE1, LEX1),
  ...,
  arg(MODIFIERm, HANDLEm, LEXm)
]).
```

The first two arguments of the predicate `ev` are the category and the lexical of the *event* (i.e. the category and

the lexical of the action accomplished by the object versus the direct object). The third argument is a set of participants (agent and direct object and modifiers), expressed as list of Prolog predicates. The category of the agent (`AGENT_CATEGORY`), the category of direct object (`DIROBJ_CATEGORY`) as well as their lexical form (`AGENT_LEX` and `DIROBJ_LEX`) are included in the predicative description of the event argument (`arg`).

Linking rules should therefore describe when two news items have to be linked together. These are written over the objective representation of the investigated documents. In particular, they exploit the notion of event. Linking rules are then Prolog predicates defining a linking criteria that motivates the existence of an link among the source and the target news items from which events are derived. Linking rules define all the constraints that the participants of two events must satisfied for generating a link between them. Each generated link has therefore a *LINK_TYPE* that is determined by the application of a specific rule. In order to compile a linking rule a list of pre-defined constraints, expressed as predicates, needs to be defined. The constraints act over the basic constituents of an event (i.e. event lexical/category, subject, object and modifiers). In particular as the event category and lexical are supposed to have a different semantic from subject, object and modifier, two type of constraints have been defined. More precisely a linking rule is a Prolog predicate of the form:

```
lrule( LINK_TYPE,
      SOURCE_EVENT_CATEGORY,
      TARGET_EVENT_CATEGORY,
      SET_OF_EVENT_CONSTRAINTS,
      SET_OF_ARGUMENT_CONSTRAINTS )
```

where:

- *LINK_TYPE*, is the type of the link that is generated by such a rule.
- *SOURCE_EV_CATEGORY* and *TARGET_EV_CATEGORY* are the category of events involved in the linking rule. For example in case of an event that relates to a meeting and another event that relates to an acquisition of stocks in that meeting, it would be useful to have a linking rule characterised by `MEETING_EVENT` as category of source event and `BUY_EVENT` as category of target event.
- *SET_OF_EVENT_CONSTRAINTS* is the set of constraints to be activated on the event category/lexical information of the source and target events.
- *SET_OF_ARGUMENT_CONSTRAINTS* is the set of constraints to be activated over the arguments of the source and target events.

Given the above description a linking rules which expresses correlation between the participants of a meeting and a company acquisition in the meeting could be:

```
lrule('Acquisition during a meeting',
      MEETING_EVENT, BUY_EVENT,
```

```
SET_OF_EVENT_CONSTRAINTS,
SET_OF_ARGUMENT_CONSTRAINTS)
```

The *SET_OF_ARG.CONSTRAINTS* specify relation between the participants of the meeting and those that acquire something. The *SET_OF_EVENT_CONSTRAINTS* specify the relation between MEETING_EVENT and BUY_EVENT as well as the lexicals associated to them.

3.2.2. Expressing constraints in the linking rules

The aims of the constraints are to select the properties of the participants and the properties of the event categories. These constraints compositionally build linking rules. A simple set of constraints is:

- *Category Identity*: two participants must be of the same category. This implies that two entity must belong to the same class. For example IBM and INTEL are both companies so they belong to the company category. If a Category identity constraint is included inside a SET_OF_EVENT_CONSTRAINTS, it casts different events to be in the same category. If we use this constraint leaving unspecified the event category we are grouping together event of the same category.
- *Lexical Identity*: the participants must have the same lexical e.g. the participant *Bill Gates* is the same lexical in *Bill Gates buy IBM* and in *Bill Gates get married*. A rule based on the category identity constraint would not be useful in the above case as a lot person get married. The Lexical Identity for the set of event constraint is less meaningful. However it can be used to specify the relation involved in a couple of events more precisely. For example if we have the event *Bill gates sell IBM*, its category will be *acquisition*. This information would not useful if we want build a rule for capturing document about company selling.
- *Conceptual Similarity*, it is an extension of the category identity type. In this case categories are grouped in a hierarchical structures. It is possible to express a relation of parents among participants.

Given the above constraints the following events:

```
ev(MEETING_EVENT, invite,[
  arg(AGENT, Company, Intel),
  arg(DIROBJ, person, Bill Gates),
  arg(MODIFIER, in, Seattle)
]).
ev(BUY_EVENT, acquire,[
  arg(AGENT, person, Bill Gates),
  arg(DIROBJ, company, Intel)
]).
```

two sample rules for capturing the link type *Acquisition during a meeting* are:

```
lrule('Acquisition during a meeting',
  MEETING_EVENT, BUY_EVENT,
  [],
  [lex_id(AGENT,DIROBJ)]
).
```

```
lrule('Acquisition during a meeting',
  MEETING_EVENT, BUY_EVENT,
  [],
  [lex_id(DIROBJ,AGENT)]
).
```

It is worth noticing that the above rule involves general events so the information about participants has to be more specific (i.e. lexical information about participants is needed). This pushes for the use of *lex_id* constraint.

Another generic rule is that groups document speaking about a target agent doing whatever action. For example the events in which Bill Gates buy something could be captured by the following rule:

```
lrule('Same participants rule',
  '--'
  [cat_id()],
  [lex_id(AGENT,AGENT)]
).
```

In the above rule the only requirement is the same agent in the linking documents. The agents in the target events have to do an action of the same category type (e.g. Acquisition event, Announce event, Market strategy events,...).

When is needed grouping together documents in which a target action is carried out, it is possible to use the category constraints for the agent and object (i.e. the *cat_id* constraint). For example a linking rule in which agents of the same category make acquisitions of object of the same category is the following:

```
lrule('Person acquire Company',
  BUY_EVENT, BUY_EVENT,
  [],
  [cat_id(AGENT,AGENT),
  cat_id(DIROBJ,DIROBJ)]
).
```

3.3. The linking algorithm

Once the linking rules formalism has been developed it is possible to design the linking algorithm. This should takes as input the ORs of two documents: the source and the target. For each couple of events in the source and in the target, the linking rule database *LRDB* is considered. If some rule is matched a link is generated and it is stored in a link DB. The rules are composed of some basic constraints that act on the constituents of an event. In this way, if an extended list of basic constraints is available it is possible for the user to define several linking rules. The rule can be described in an external data file so new rules can be added to the similarity model without re-designing the entire architecture.

The linking algorithm takes as input two documents, one is the source *S* and the second is the target *T*, and given their sets of events, respectively *Ev(S)* and *Ev(T)*, check if any couple $\langle ES, ET \rangle$, where $ES \in Ev(S)$ and $ET \in Ev(T)$, satisfy any of the linking rules contained in *LRDB*.

The algorithm is composed of the following steps:

```

function Link(text S, text T) returns Linkset
begin
  Linkset  $L = \emptyset$ ;
   $Ev(S) = BuildEv(S)$ ;
   $Ev(T) = BuildEv(T)$ ;
  for each  $(ES, ET) \in Ev(S) \times Ev(T)$ 
    begin
      while  $((R = SelectNextRuleFor(ES, ET)) \neq \text{NULL})$ 
        begin
           $R = (RuleType, SEvCat, TEvCat, CatConstr, ArgConstr)$ ;
          if  $(ApplyCatConstr(CatConstr, ES, ET) == \text{true})$ 
            begin
              boolean sat = true;
              while  $(SArg, TArg) \in nextArg(ES, ET) \text{ AND } \text{sat}$ 
                sat =  $ApplyArgConstr(ArgConstr, SArg, TArg)$ ;
                if (sat)
                   $AddLink(ES, ET, RuleType, L)$ 
            end
          end
        end
      end
    end
  return L;
end

```

4. Conclusions and future work

In this paper, we have presented a methodology for the automatic hyper-linking among news items. The presented approach is based on Information Extraction techniques that give the possibility of building semantically motivated links among documents. This approach is more expressive than the traditional approaches to the problem that allows the automatic construction of links only between related documents. The approach has been used to build the Namic prototype (EU-founded project NAMIC, News Agencies Multilingual Information Categorization, IST-99 12392).

As the approach is rather different from the pre-existing the comparison is hard. We will therefore compile, according to our definition of the task, a large test set that should enable the validation of the methodology and of the implemented system.

5. References

- James Allan. 1995. Automatic hypertext construction. Technical Report TR95-1484, 13.
- Roberto Basili, Maria Teresa Pazienza, and Michele Vindigni. 2000a. Corpus-driven learning of event recognition rules. In *Proc. of the Workshop on Machine Learning for Information Extraction, held jointly with ECAI2000*, Berlin, Germany.
- Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. 2000b. Customizable modular lexicalized parsing. In *Proc. of the 6th International Workshop on Parsing Technology, IWPT2000*, Trento, Italy.
- Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. 2002. Learning ie patterns: a terminological perspective. In *Proc. of the Workshop on Event Modelling for Multilingual Document Linking, held jointly with 3rd LREC*, Canary Islands, Spain.
- David Ellis, Jonathan FurnerHines, and Peter Willett. April 1994. The creation of hypertext linkages in fulltext documents: Parts i and ii. Technical Report RDD/G/142.
- Robert Gaizauskas and Kevin Humphreys. 1997. Using a semantic network for information extraction. *Natural Language Engineering*, 3, Parts 2 & 3:147–169.
- S. Green. 1997. *Automatically generating hypertext by computing semantic similarity*. Ph.D. thesis, Department of Computer Science, University of Toronto.
- Steve Outing. 1996. Newspapers online: The latest statistics. *AEditor and Publisher Interactive [Online]*.
- Randall H. Trigg. 1983. *A networkbased approach to text handling for the online scientific community*. Ph.D. thesis, University of Maryland.
- J. Christopher Westland. 1991. Economic constraints in hypertext. *Journal of the American Society for Information Science*.

Ontology-based author profiling of documents

Jan De Bo, Mustafa Jarrar, Ben Majer, Robert Meersman

VUB STARLab
Vrije Universiteit Brussel
Pleinlaan 2
Brussels Belgium
{jdebo, mjarrar, bmajer, meersman}@vub.ac.be
Telephone:+32 2 6293487 Fax:+32 2 6293525
Home Page: <http://www.starlab.vub.ac.be/>

Abstract

In this paper we present the advantages of using an ontology service for the modelling of user profiles in the EC FP5 IST project NAMIC (IST-1999-12392). By means of an ontology server people set up user profiles, which are in fact views, i.e. *specifications of queries* on the ontology. These views are constructed using a JAVA API, which forms the *commitment layer* of the ontology, built on top of an ontology base. In NAMIC an ontology server is used to establish a link between the lexical object representations, generated by the natural language processors (NLP) on the one hand and the user's interest, specified through the selection of relevant concepts and facts of the ontology on the other. This allows to specify a user profile independently of language, categorization and NLP specific "world models". Users then set up a profile consisting of events, agents participating in these events and other content information in which they are interested in. For instance, a journalist writing articles about financial issues may be interested in related documents containing a "raise event" of company shares. If he has specified those conditions in his profile he will be able to retrieve resources which contain events that are semantically related to that kind of event pattern. User profiles in NAMIC do not have to be static. The results of processing by the NLPs of a document the user is currently working on, may be used to construct a dynamic profile, which may contain events specific for that document. This way a user's profile can be dynamically adapted to his current interests. We also developed a tool which illustrates the creation of user profiles using ontological concepts and facts.

1. Introduction and Motivation

In this paper we present results derived from our work in the NAMIC project. Within the NAMIC project the main objective was to develop advanced technologies of Natural Language Processing for multilingual news customization and broadcasting throughout distributed services, which represents one of the major problems for International and National News Agencies (NA) as well as for the spread of Web technologies. Within their own business cases, NAs need to integrate in their own repositories news distributed by other NAs usually in different languages and according to different classification standards. Mismatching is at language level, since different languages are used, as well as at the conceptual, as the organization/storage of news proceeds according to diverging schemes. The volume and richness of this information has, however, a catch: it can overwhelm the pressed user-journalist that may be looking for a particular type(s) of information. This is a well-known problem in an information-rich environment, and especially in the case of (large) sets of hyperlinked documents, often referred to as the "lost-in-hyperspace syndrome".

Several aspects have been researched to improve searching, browsing and retrieval of information. In the information retrieval approach, several techniques ranging from string matching to advanced lexical analyses systems are used in order to understand the implicit semantics and thus the relevancy of the data that will be retrieved. On the other side, in the artificial intelligence and database approaches, such as for example the semantic web, the semantics (and the syntax) of the data are explicitly defined and linked with knowledge bases as ontologies, which help to make precise queries or for reasoning. Experience shows that the accuracy of extracting the

implicit semantics and the relevancy of the data is low, e.g. a search using regular search engines results in a huge amount of information, especially for large volume information resources such as the web, expanding queries to improve recall may also cause huge result sets. On the other hand, defining the semantics of the information explicitly, and reasoning about them in order to retrieve relevant information is an expensive task, and the scalability is very low. Therefore, we believe that combinations of these two approaches will be very fruitful for the improvement of information retrieval, as will be argued in the next sections of this paper.

Within the NAMIC project the User Domain Profiling System (UDPS) allows defining of user profiles for the filtering of news streams according to the specific interests of a user which for NAMIC, primarily would be journalists or other text writers. These user profiles are then used to exclude irrelevant items from a constant stream of documents before these documents are presented to the user.

As will be argued later in this paper, the use of an ontology has critical improvements: IR systems will gain from ontologies richer knowledge representation and modelling capabilities, improved recall by expanding the queries according to well-defined and consistent relationships in the ontology and improved precision by allowing the definition of personalised profile systems as queries against (an) ontologie(s) in order to include or exclude (a) certain type(s) of information.

Structure of the paper. In section 2 we give an introduction of what an ontology is and its critical added value for NLP based systems. Section 3 then gives the definition of a user profile and explains more details about the advantages of using ontology-based information filtering systems such as user profiles. Section 4 demonstrates the implementation done in the Namic Project and Section 5 draws preliminary conclusions and

maps ongoing and future work. Section 6 then places all acknowledgements.

2. Using ontology with NLPs

In this section we will illustrate the advantages of ontologies and their potential role in several aspects of information retrieval and how they can be used in defining user profiles.

Ontology¹ in computer science is a branch of knowledge engineering, where agreed semantics of a certain domain are represented formally in a computer resource, which then enables sharing of information and interoperation between systems. Representing the semantics (as a formal interpretation) of a certain domain implies the conceptualisation of the domain objects and their interrelationships in a declarative way, so that they can be processed, shared, and reused among different applications. Note that an ontology is more than a taxonomy or classification of terms, since it includes richer relationships between terms, e.g. “part-of, location-of, value-of, synonym-of...”(Figure 1). An ontology provides a higher level of knowledge², where the ontology terms are chosen carefully, consistently, and with a higher level of abstraction.

In the DOGMA model described summarily below, we separate relevant ontological relationship knowledge as set extensions of context-specific binary fact types called *lexons*. These express (within this assumed context) plausible relationships between concepts, using lexical terms in a given language; we implicitly assume that these terms are aligned with a lexicon (“terminology base”) that is agreed among all users of the ontology (Jarrar, 2002).

Example. The following –very partial ontology (Tables 1,2,3)- could be lexons in some arbitrary hopefully self-understood syntax, the format for the purpose of textual illustration being (#*contextid*) <term1>[<role label><term2>]; details or omitted in this paper. The ontology base, which contains the set of lexons of the modelled domain, is also known by the symbol, Ω .

(# <i>my_company-ID</i>) employee
is_a person
has first_name
has last_name
has empl-id
has birth date
has salary
works_in department

Table 1

(# <i>my_company-ID</i>)salary
is_a salary
reviewed_in month

Table 2

(# <i>employment-ID</i>) salary
has amount in-\$
expressed_in currency
converted_to currency
earned_by employee

Table 3

Through the use of ontologies one is able to express semantic relations between terms, rather than is the case with ordinary categorisations. To express these meaningful relations between different terms we need advanced modelling methodologies, like the ORM conceptual modelling language. We chose ORM for its rich constraint vocabulary and well-defined semantics. Within STARLab we also developed an XML-based ORM markup language (ORM-ML) as a means of exchanging data semantics between different agents. (Demey et al, 2002)

The enormous growth of the Web causes search engines to return a large number of pages to the user for a single search. It is time consuming for the user to traverse the list of pages just to find the relevant information. We claim that information filtering systems based on ontologies will assist the user by filtering the data stream and delivering more *relevant* information to the user. Below are a few examples of how this can be achieved. We will discuss these topics in section 3 in more detail.

IR will benefit from ontologies more than terminology bases/resources since the knowledge is more formally represented than in term bases, which facilitates the representation, maintenance, and dissemination of terminological data and makes these data reusable by computer systems in various applications. Recall and precision of search operations will be improved using ontologies to model the knowledge contained in a system. Recall will be improved by exploiting the rich structure of an ontology and specifying generic queries (Guarino, 1999). The semantics in an ontology makes it quite attractive for query expansion, because there is a strong need to expand queries with relevant terms and meaningful relations which contain a lot of semantics, for instance to include subtopics or to personalize the query according to a user’s personal interests. Precision will be increased through the disambiguation of terms and the ability to navigate through the ontology for the selection of more specific queries (Guarino, 1999).

While ontologies offer highly advanced modelling capabilities our experience indicates that, in the domain of Natural Language Processors (NLPs), ontologies will mostly be lighter, and therefore less expressive, than in other applications such as for example reasoning systems where the reasoning rules (defined as a logical theory in the *commitment layer*; containing for example the following constraint ORM.Mandatory(employee has_birth date)) are the most important part of the ontology, while NLP applications may see the lexons in the ontology base as canonically and linguistically structured expressions.

Furthermore, the context will provide added value to disambiguate (or approximate) the meaning of terms and relations.

Usage of an ontology also offers advantages for multilingual Information Retrieval. Since the ontology is a shared agreement about a (abstract) conceptualization it is in principle independent of a particular natural language

¹ In philosophy, Aristotle defined ontology as the science of being.

² The Knowledge Level is a level of description of the knowledge of an agent that is independent of the symbol-level representation used internally by the agent, (Gruber, 1995)

inherit from this concept and ignores all parent concepts (assuming the relation between the concepts is SubClassOf).

The ontology is separated from the objective representations used by the natural language processors. Since the user profile is a query on the ontology, this separation hides the user from the potentially large amount of objective representations used by the NLPs. The advantage of the independence between the underlying objective representations and the user setting up his profile is that he does not have to be aware of the different objective representations of the NLPs. The ontology can thus be seen as an intermediate level shielding the different representations of the NLPs from the user. Once the ontology is built, natural language processors will have to adapt their objective representations to it. This way a query on the ontology, can be considered to interact independently with the objective representations generated by various natural language processors.

Because of the multilingual data resources, development of different natural language processors (in NAMIC, English, Spanish and Italian) is required. This was done by the universities of Sheffield, Rome (Tor Vergata) and Catalonia (Universitat Politècnica de Catalunya). The user profiling system, introduced in NAMIC, however enables the user to specify language-independent queries, but still gives the possibility to get back related documents in all languages provided by the news agencies.

As mentioned before a user has the possibility to specify his interests in a static profile by selecting the appropriate relations and concepts from the ontology. It is however quite possible that a journalist's interests change while working on a particular news story. Therefore the user has to adapt his profile according to his current needs and interests instead of having to create an other additional profile. User profiles, developed within the NAMIC project, can be dynamically adapted. Indeed, as part of the NAMIC profile services, a journalist has the possibility to create a local profile according to the text he is currently working at, because it is likely that he will be interested in retrieving documents containing events, or knowledge related to agents participating in events which he has already entered in his text. The user is given the possibility to update his current static profile according to this new profile, making his own profile change dynamically. This prevents the user from having to manually annotate his own article of text by adding (ontologically derived) concepts and relations to his static profile, assumedly saving time and improving consistency.

4. Implementation

The ontology service in NAMIC provides the possibility to store, edit and retrieve ontological information that models (partial) semantics relevant to the project's domain and in particular the ability to define user profiles based on these semantics.

In order to satisfy the requirements mentioned above we developed a tool, with the following classical two-tier client/server architecture, illustrated in Figure 2.

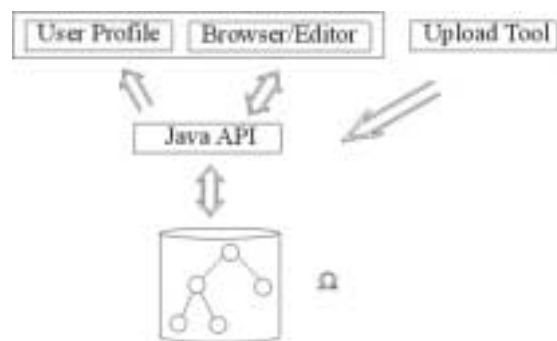


Figure 2

- At the bottom of Figure 2, there is a storage facility for the ontology (in a database)
- Above that, an intermediate API layer establishes communication between various tools and the ontology.
- At the top, support tools like browsers, editors and user profiles are implemented.

In our paper we will use the term 'objective representations' of the natural language processors to refer to Event Matching patterns, which are described in detail in (Basili et al). The process of ontology engineering begins with the development of a base model that provides a framework for the integration of other different, individual resources. The creation of this ontology base can be viewed as a conceptual modelling task, based on ontology merging and alignment of the available resources. The result contains the fundamental concepts based upon the natural language processors' objective representations, that are generally useful for the project. For instance, consider the following verb syntactic frame: 'person - sells - attribute' as an example of an objective representation from the NLPs' event matching rules. The verb syntactic frame which is not considered to be an ontological concept, is mapped to 'Company Acquisition event'. The occurrence of this verb syntactic frame in a document then results in the detection of a 'Company Acquisition event'.

The individual resources that are considered for their incorporation into the NAMIC ontology were the following:

- The IPTC category system (IPTC)
- The EuroWordNet base concepts (EuroWordNet toplevel concepts) (Vossen, 1998)
- Named Entity lists (Stevenson et al)
- Event Types (Basili et al)

In order to integrate the natural language processors' objective representations of the different individual resources into the ontology, an alignment process needed to be performed between those different representations. Categories, events and named entities are aligned with EuroWordNet base concepts, by establishing mappings between the involved concepts of the different resources considered for integration in the ontology. This is illustrated in Figure 3; the alignment mappings are depicted as double-sided arrows.

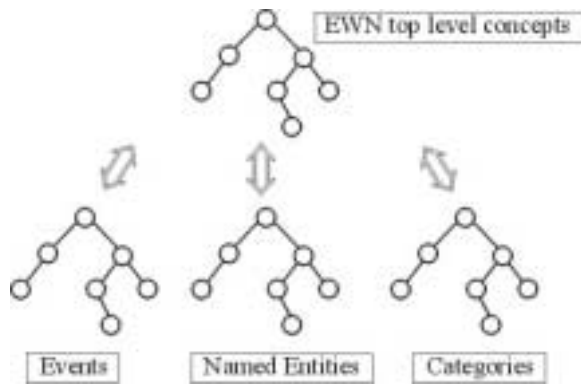


Figure.3

Because an ontology is a *shared* agreement about (a conceptualisation of) the world, aligning different ontologies with one another is required in order to obtain agreement between the concepts of the different ontologies. In order to develop tools automating this activity, good context formalisms will undoubtedly become helpful here but within the scope of NAMIC we had to align the different ontological concepts manually. At this state of the art it is as yet unrealistic to expect that merging or alignment at the semantic level could be performed completely automatically. A prototype of a tool to assist ontology merging and alignment has been built by the Stanford Medical Informatics department of Stanford University. This tool, based on the SMART algorithm, is an extension of the Protégé (Noy, 1999) ontology-development environment.

For the purposes of NAMIC we have also developed a simple custom tool (OntoNAMIC) to make the ontology available for browsing, editing and setting up user profiles.

The browser window consists out of a left pane and a right pane. The left pane is responsible for browsing through the ontology, while the content appearing in the right pane depends on whether one has selected the class view, diagram view or profile view on the toolbar of the application.

When the domain expert (i.e. typically *not* the journalist) selects the Classview, all the lexons containing the selected concept on the left will be displayed in the right pane. Choosing the Diagram view enables one to drag and drop concepts from the left pane into the right.

By double-clicking on this dropped concept an ORM diagram appears, displaying all the lexons of which the concept is a part. ORM is a well-known conceptual modelling language (Halpin, 2001) here "re-used" (in part; some interesting modifications are needed that however will be the subject of a separate paper) to represent part of the ontology. In the diagram, ovals represent *entity types*, the rectangles are arbitrary (uninterpreted) relationships between them, and arrows are (interpreted) *is-a relations*. The important point is that it is possible to map such models to and from lexon-based ontologies, which provides two immediate benefits: a graphical and formally founded notation, and existing tools that already support it, such as Microsoft's VisioModeler for ORM. Because of our earlier experience with this particular method and tools for database design (De Troyer et al, 1995), we have adopted it as a prototypical research and implementation tools and techniques environment for ontology construction.

One then sets up a user profile by choosing the profile view on the toolbar. Remember a user expresses his interests in his profile by specifying a query on the ontology, i.e. as a composition of logical combinations of the desired events, EWN concepts, named entities and categories from the ontology. The resulting implied logical expression will then specify which documents satisfy the profile. This is illustrated in Figure 4.

5. Future work

Although we have now chosen to use a rather simple query language for setting up the user profiles, it is our aim for future work to develop a more sophisticated conceptual query language (for instance similar to RIDL (Verheyen et al,1982)), to specify queries on the ontology.

6. Acknowledgements

This work was supported by the European Commission's IST Project NAMIC (IST-1999-12392). We would also like to acknowledge contributions by our partners in this project Agenzia ANSA S.C.R.A.L., The University of Sheffield, University of Roma Tor Vergata, Universitat Politècnica de Catalunya, Vrije Universiteit Brussel, Comité International des Télécommunications de Presse, Itaca s.r.l., Agencia EFE, S.A. and Financial Times.

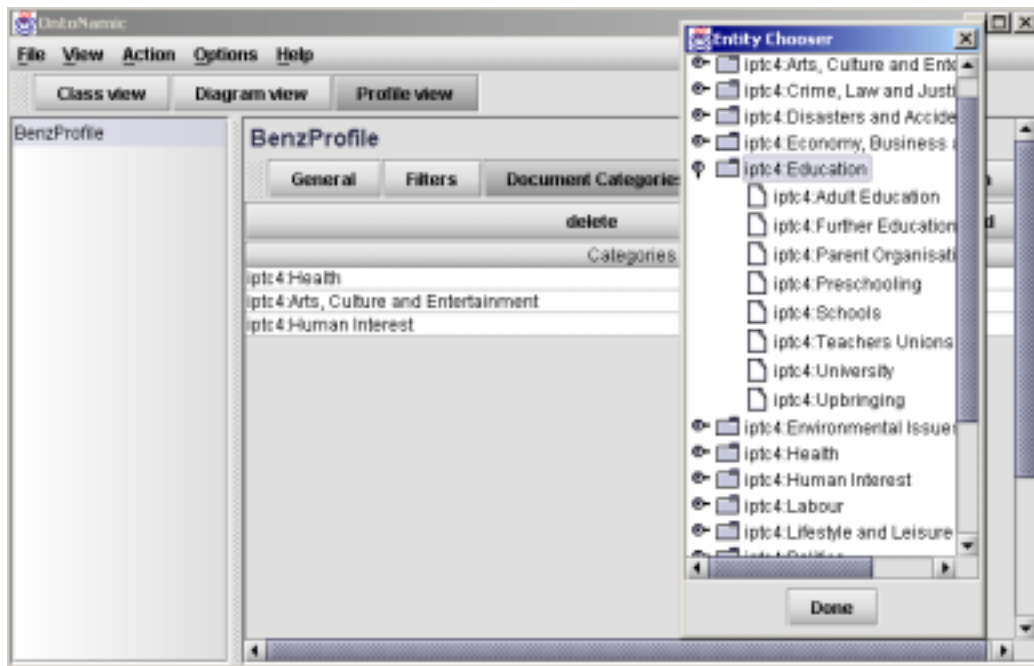


Figure 4

7. References

- Abuzir Y. and Vandamme F: "E-Newspaper Classification and Distribution Based on user profiles and Thesaurus", SSGRR 2002w - International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet, January 21 - 27, 2002 L'Aquila (Italy).
- Abuzir Y., Vervenne D., Kaczmarek D. and Vandamme F.: "E-mail messages classification and user profiling by the use of semantic thesauri", in CIDE 2001 - in Proceedings CIDE 2001 Conference - 4th International Conference on the Electronic Document, Toulouse - FRANCE Oct. 2001.
- R. Basili, R. Catizone, L. Padro, M.T. Paziienza, G. Rigau, A. Setzer and N. Webb, Y. Wilks and F. Zanzotto, 2001 Multilingual Authoring: the NAMIC approach, Human Language Technology and Knowledge Management, EACL/ACL Workshop, Toulouse, France
- Demey J., Jarrar M., Meersman R., Exchanging ORM Schemas Using a conceptual Markup Language, Submitted to ER2002
- De Troyer, O., Meersman, R., 1995 : "A logic Framework for a Semantics of Object Oriented Data Modeling" , in: Proceedings of Entity Relationship and OO Modelling Conference, Papazoglou et al. (eds.) Springer LNCS.
- Guarino, N., Masolo, C., and Vetere, G. 1999. OntoSeek: Content-Based Access to the Web. IEEE Intelligent Systems, 14(3): 70-80.
- Gruber T., 1995 "Toward principles for the design of ontologies used for knowledge sharing", International Journal of Human-Computer Studies, 43(5/6).
- IPTC, <http://www.iptc.org/> -> Subjects -> Subject reference system
- Jarrar M., Meersman R., 2002 Practical Ontologies and their Interpretations in Applications - the DOGMA Experiment, Submitted to WWW02
- Karp, P. 1992. The Design Space of Frame Knowledge Representation Systems. Technical Report 520, SRI International Artificial Intelligence Center
- Noy, N.F., and Musen, M.A. (1999). SMART: Automated Support for Ontology Merging and Alignment. Submitted to the Twelfth Workshop on Knowledge Acquisition, Modeling, and Management, 1999. Banff, Canada
- Pretschner, A., and Gauch, S. 1999. *Ontology based personalized search*. In Proc. 11th IEEE Intl. Conf. on Tools with Artificial Intelligence, pp. 391--398
- Stevenson, M. and Gaizauskas R: Using Corpus-derived Name Lists for Named Entity Recognition.; Applied Natural Language Processing and the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL-2000) Seattle, WA
- Terry Halpin : Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. Morgan Kaufmann Publishers, 2001. ISBN 1-55860-672-6
- Verheyen, G. and van Bekkum, P.: "NIAM, aN Information Analysis Method", in: IFIP Conference on Comparative Review of Information Systems Methodologies, T.W. Olle, H. Sol, and A. Verrijn-Stuart (eds.), North-Holland (1982).
- Vossen P (eds), 1998; EuroWordNet: A Multilingual Database with Lexical Semantic Networks, Kluwer Academic Publishers, Dordrecht (Netherlands)

Description of Events: An Analysis of Keywords and Indexical Names

Khurshid Ahmad, Paulo C F de Oliveira, Pensiri Manomaisupat, Matthew Casey and Tugba Taskaya

Department of Computing, University of Surrey
Guildford, Surrey. GU2 7XH
UK
(k.ahmad@surrey.ac.uk)

Abstract

Event modelling requires a good understanding of the modes used in communicating the events, including natural language, graphs and images. A case study of financial market movement, where text, or news wires, and graphical information, or a financial time series, were correlated, is described. This leads to a need for automatic text classification: a method based on unsupervised neural networks and autonomous assignment of keywords is described. These are preliminary results of an EU 5th Framework Project –GIDA (No. IST 2000-31123). Methods of corpus linguistics and terminology are used to underpin the methods.

1. Introduction

An event is defined as a significant occurrence or happening, or more specifically, as in physics, an event is a phenomenon or occurrence located at a single point in space time. In the late 20th century a tautological compound *news event* makes the meaning of the word ‘event’ even more explicit. A description of an event names the persons, places, things, or ideas in relation to the significant occurrence, happening or phenomenon: Osama bin Laden is frequently named in relation to terrorism; the Financial Times/Stock Exchange 100 (top companies) index (FTSE) is named in relation to the British, and possibly, EU economy; *relativity* and Einstein in relation to 20th century physics.

Reports of terrorism, stock market movements, and developments in theoretical physics, use written language, photographs, time series of financial transactions, graphs of key variables, and other symbol systems. Reports of events, political, economic, scientific or leisure, for instance, are crafted using a range of semiotic systems – from natural language to images, from time series to icons. An event, when described in natural language, involves the deliberate frequent use, and at times deliberate censoring of names related to the significant occurrence or phenomenon. For a specific event, described over a period of time in a number of texts, some persons, things or ideas are mentioned more or less frequently depending on their influence on the event. An event, perhaps at the lexical level of linguistic description, is a cluster of keywords or terms related to the specific area of human activity – terrorism, finance and commerce, physics, or football for example.

The names of (significant) persons, things or ideas act as an index to an event, an index which has linguistic rendering but can equally be referred through the use of other semes – images, graphs, mathematical symbols, circuit diagrams are some of the other indexical semes. Keywords-in-context (KWIC), largely common nouns sometimes qualified by adjectives, can be used to categorise documents related to a special subjects or, perhaps indirectly, to a specific events.

For us, event modelling requires an understanding of keywords and a collation of indexical names. For computer-based event modelling, involving information

extraction and retrieval, and text understanding, it is important (a) to automatically identify and verify new keywords and indexical names, (b) to be able to note nuances of, and changes in, use of the keywords and the indexical names, and (c) to correlate the information in text and in graphs through the use of indexical names and keywords.

News streams provided by organisations like Reuters or Bloomberg comprise a range of keywords and indexical names that may change from one news item to the next; an event modeller will need to filter the news from such a diverse information resource. Specialist information providers deliver not only news texts but also supply, for example, time series of changes in value of stocks, shares, currencies, bonds and other financial instruments.

We have a narrower focus than other authors in information extraction (see for example Gaizauskas et al, 1995 and Maybury et al, 1995) in that we are looking for changes in key financial instruments that are reported in financial news-wires. The news coverage of these instruments is of two types: first, there is a daily report about changes in the value (numerical) of the instruments for instance, one can see time series comprising historic data about the changes in values of currencies; second, the manner in which the value of the instruments changes depends on the reports relating, directly or indirectly, to the instrument. The reports, for example, about war or economic uplift/downturn, affect the value of the instruments. Some authors claim that there is a correlation between ‘good’ or ‘bad’ news relating to the instrument and its potential numerical value. In Section 2 we take this discussion further.

The news report is one of the most commonly occurring linguistic expressions. Despite being a good example of open-world data, a news report is a contrived artefact: each report has a potentially attention grabbing headline; the opening few sentences generally comprise a good summary of the contents of the report; there are *slots* for the date of origin and slots for photographs and other graphic material. This contrived artefact is highly focused and highly perishable, and usually contains references to one or more persons, places, events or actions. Automatic categorisation of news stories is of substantial interest to in a range of applications (Mani 1998) to information retrieval communities, and to major news vendors

supplying *on-line news*; Section 3 takes up this story further and we conclude in Section 4.

2. Keyword and Indexical Name Correlation

Generally, information is delivered to financial market operatives via electronic mail, newspaper, or company announcement briefings or company annual reports. Whatever its source, the information in the news is an important component in making investment decisions (Figure 1). Equally important are events like natural disasters or terrorist activities for example.

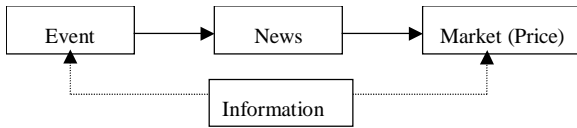


Figure 1: The relationship between Events, News and Markets (price) through Information.

For example, the terrible events of September 11, 2001, have had a catastrophic effect on financial markets world wide (See Figure 2). Various national economic indicators –indexical names – show the reaction on the date; there has been a decline in the value of these indices before that date and indeed a resurgence in the value afterwards.

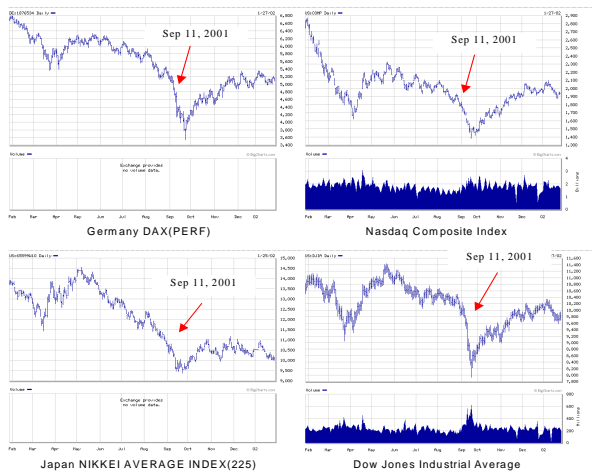


Figure 2: Movement from Feb 2001 to Jan 2002. Note the dip on and around Sep 11th 2001.

According to the Dow-Hamilton Theory (Rhea 1994), there are three kinds of price movements or market movements: (i) *Primary movement* which lasts from few months to many years and represents the broad trend within a market; (ii) *Secondary movement* last from a few weeks to few months and may sometimes be contrary to the primary movement; and, (iii) *Daily Fluctuations* can move with or against the primary trend and exist for a few hours to a few days.

2.1. Market Movement and Market Sentiment

Our work, sponsored in part by the EU-sponsored GIDA project (Project No. IST 2000-31123), focuses on primary movements. We report on some initial work that attempts to changes in an index, FTSE100, with changes in ‘market sentiment’ as expressed in news reports about the UK economy specifically and reports about the Wall Street indices. The later has substantial influence on the

UK economy. Financial analysts use sophisticated political, economic and psychological analysis to determine the reaction of market operatives and to predict the possible trading decisions of the operatives. Reports related to the sentiment use a range of metaphors to express the state of a market and its possible movements. Francis Knowles has written about the use of *health metaphors* used in the financial news reports: markets are full of *vigour* and are *strong* or the markets are *anaemic* or are *weak* (1996); most newspapers also use *animal metaphors* – there are *bull* markets and *bear* markets, the former refer to expansion, and indirectly to fertility, and the later to shy, retiring and grizzly behaviour much like that reported about bears in popular press and in literature for children. Indeed, there are fairly literal words that express the sentiment, as reported in the news wires, about the markets: financial instruments *rise*, *fall*, markets *boom*, *go bust*, and there are *gains*, *losses* within the markets, economies *slowdown*, suffer *downturns*, whole industry sectors maybe *hardpressed*. Table 1 contains examples of good and bad news in a typical Reuters news stream:

Mainly Good News Stories	Rather Bad News Stories
Naval shipbuilder and military contractor Vosper Thornycroft has boosted its civil arm by buying facilities manager Merlin Communications (Nov 14, 2001)	Heavyweight banking and oil stocks have dropped up the leading share index as investors bet on fresh interest rate cuts.’ (Nov 21, 2001).
The FTSE 100 stock index looks set to open stronger today after Wall Street added to gains seen at the London close and with U.S. stock index futures boosted by rumours that Osama bin Laden had been captured.’ (Nov 15, 2001).	The European Commission has slashed its official growth forecasts for the euro zone [...], predicting the most serious slowdown since the 1990s recession, with lower growth in 2002 than this year.’ (Nov 21, 2001).
Builder McCarthy & Stone has posted a 13 percent rise in annual pre-tax profits, built on strong sale prices for its retirement homes [...], but cautions that the boom may be over.(Nov 15 2001).	The FTSE 100 fell today, amid concern about how the U.S. economic downturn will hurt technology stocks and British Airways’ operations. (Dec 10, 2001).
Leading shares are expected to rise again after Wall Street steamed higher overnight and the market basked in a feel-good glow, dealers said.’ Nov 14, 2001).	Britain’s economy appears to be sailing along relatively smoothly despite the global slowdown and a string of high-profile job layoffs (Oct 22, 2001).
‘Leading shares have edged higher in early trade, boosted by gains in technology stocks in response to a Wall Street rally and positive expectations for the economic outlook.’ (Jan 4, 2002).	‘The hard-pressed manufacturing sector has recorded its biggest monthly production drop in almost a decade, sinking deeper into recession . (Nov 5, 2001).

Table 1. Examples of ‘good’ and ‘bad’ news stories in Reuters News Wires (Oct 2001-January 2002)

The above table contains examples of how the market is moving. But here we have free natural language complete with ambiguity and nuances of meaning: so there maybe a ‘rise in profits’ and a ‘strong sale prices’, in the story about builder’s McCarthy & Stone above, both phrases suggesting that this is a good news story, except for the last sentence suggesting that ‘boom maybe over’. Nevertheless, many of the news items do not change the nuance of the story by such highly temperate notes.

2.2. Correlating Sentiment and Market Indices

We created a corpus of 1,539 English financial texts from one source (Reuters) on the World Wide Web, published during a 3 month period (Oct 2001-January 2002) comprising over 310,000 tokens. The corpus comprised a blend of both short news stories and financial reports. Most of the news is business news from Britain with thirty percent of the news is from Europe and from the United States.

We found over 70 terms each for conveying good news and bad news in the above corpus. The texts in our corpus were also time stamped, and by using our text and terminology management system, System Quirk, we computed the cumulative weekly frequency of *good* words and *bad* words during one month – November 2001. The ‘week’ is a working week comprising 5 days, Mondays-Fridays:

Time (5 day Week)	Good Word Frequency	Bad Word Frequency
1	<u>58</u>	40
2	71	75
3	77	66
4	73	59
5	72	<u>28</u>
Total	351	268

Table 2: Frequency of Good and Bad words in Nov 2001. The underlined figures in the 2nd and 3rd columns indicate the minimum value of the frequency and the numbers in italics are the maximum value.

Table 2 shows that in November the highest frequency of ‘good’ words was in week 3 (77 instances) and the ‘bad’ words was in week 2 (75 instances). How does this correlate with the movements of the London stocks and shares as expressed by the FTSE 100? Figure 3 provides an example of the correlation between the frequency of ‘good’ words from news in November in our corpus and close prices of FTSE100 Index for the whole month of November.

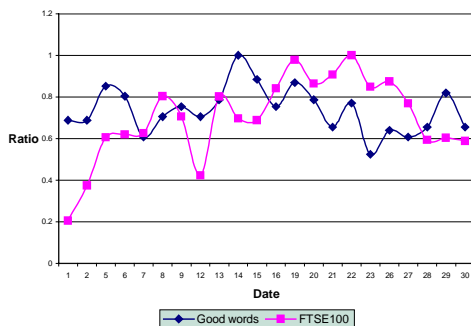


Figure 3: Market correlation between ‘good’ word frequency and FTSE index.

The highest value of the FTSE 100 index was on on 22 November 2001 (5345.94). There is a perhaps a correlation between the changes in the value of the index and the frequency of ‘good’ words: Positive gradient in

the ‘good’ words time series correlates well with the positive gradient in the FTSE 100 index values. What will be interesting for the purposes of predicting the movement of the market, will be a correlation that suggests that a rise in the number of good words one day nudges the market. Correspondingly, that a decrease in the number the previous day will lead either to a static market or falling market the next day. The same can be said, perhaps in reverse, about the bad news words (see Figure 4).

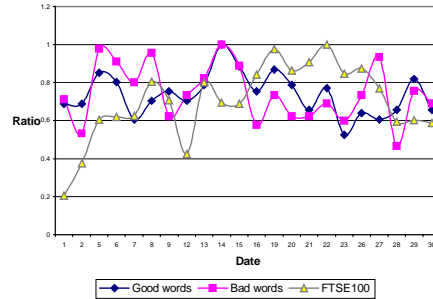


Figure 4: Good and bad word frequency correlated with FTSE 100.

Figure 4 shows ‘good’ word and ‘bad’ word frequency is perhaps correlated with FTSE100 values. For example, from 23rd to 29th November the frequency of ‘bad’ words increased while the FTSE100 went down over this period. After the 29th November, the FTSE100 value slightly increased as the ‘good’ word frequency also increased.

The above analysis and the concomitant results are of a tentative nature in that work is progressing in three major directions. First, one needs a bigger corpus, and a longer time series, to be more assertive about a correlation between an index and the corresponding sentiment-bearing terms. Second, further analysis is underway to note that the good news is sometimes tempered with bad news and vice versa – this will involve a phrasal or sentential analysis. Third, the notion of a ‘time series’ is a carefully defined concept for a series of cardinal numbers collected at discrete intervals of time or collected continuously; we are exploring the status of a time series made up of counts of lexical strings found in a news report that may have been produced over an approximate time. Nevertheless, Figures 3 and 4 show how a news stream, comprising subject specific texts, can be visualised especially in the context other index names.

3. Classifying News Wires Using Keywords

3.1. Categories of News Reports

A news stream comprises news stories that: (a) range over a whole range of subjects; (b) the news may emanate from or maybe about a nation state; and (c) the news may be focused on a certain specific area of human enterprise Reuters labels for items (a)-(c) are ‘Topic’, ‘Country’ and ‘Industry’; these labels are used by Reuters’ sub-editors to tag each news story with one or more Topic and Country tags, and in some cases with the Industry tags. These pre-assigned tags, about 1000 different tags in all, can, in principle, be used to categorise individual news stories in a news stream. However, the plurality of tags, that is the presence of one or more tags with either Topic, Country,

or Industry, makes such a categorisation more complex. Before we discuss how to deal with such a complex categorization task (see Section 3.5), which is possibly subjective in that the categories are based on an ontology which was created by Reuters themselves, we look at how to categorize texts based on (semi-)automatically extracted keywords.

One well-recognized way of describing news reports is to classify the texts as a distinct *register* or *genre* of writing. The term register is used to indicate that the language within a specialized field differs from that of *general language* or language of everyday use, at lexical, syntactic and semantic levels. A large collection of general language text may thus be contrasted with a set of specialist reports at various linguistic levels, including lexical and semantic.

An important use of this contrast is in a method of semi-automatically identifying the terms of a set of specialist domains. This method involves comparing the frequency of systematic terms in a collection of specialist texts sometimes called a *corpus*, with the frequency (or absence) of the terms in a carefully compiled corpus of general language texts. Each term can be construed as a dimension in a vector space and the presence or absence of a term within a text is then used to allocate the text its position within the vector space. There is some evidence from work in linguistics that word categories (nouns, verbs, adjectives, adverbs, prepositions, etc.) may be inferred from the statistical occurrences of words in different contexts. For Kohonen and his colleagues, “*context patterns*” consist of groups of contiguous symbols; the authors cite pairs or triplets of words in a sentence as an example of such patterns. Such *pairs* or *triplets* are then used as inputs in the training and testing of a neural network (the so-called self-organising feature maps or SOFM; details of this map is presented in the next Section 3.2). Kohonen has shown that a SOFM-trained word context pairs, derived from 10,000 random sentences, shows ‘a meaningful geometric order of the various word categories’. A larger SOFM, the WEBSOM has been variously described by Kohonen as a *scheme*, *content-addressable memory*, *method* and *architecture*. WEBSOM is a two-level self-organising feature map comprising a word category map and a document category map, which has been used to classify newsgroup discussions, full-text data and articles in scientific journals (Kohonen 1997b, Kaski *et al.* 1996). Terms were pre-selected by the builders of WEBSOM. There are other neural network architectures that have been used in text categorisation, especially the widely-used supervised learning algorithms – SOFM is based on unsupervised learning algorithm – which have been discussed by Lewis (1995).

Consider a set of texts that may have been selected according to certain criteria: for instance, all texts streaming along a news wire over a short period of time comprising news related to specialist topics – like environmental news or economic news. Such a short news stream may contain may result in a text collection, or if collected systematically, a text corpus, that may be characterised the high frequency of environment – or economics – related terms. However, over a long period of time this may not be the case as the news stream may start to deliver texts in different specialist areas. So how do we extract terms from such a corpus?

Specialist texts can be distinguished from a general language text at the lexical level of linguistic descriptions by looking at the ratio of relative frequency of a linguistic token in a specialist text and its frequency in general language texts. This ratio has been termed *weirdness* to indicate how it measures the preponderance of words in specialist texts that would be unusual in general language, (see, for example, Ahmad 1995).

Typically, before text documents are represented as vectors in order to act as the input to a text categorisation system, pre-processing takes the form of filters to remove words ‘low in content’ from the text (see the WEBSOM method in Kaski *et al.* 1996). We remove punctuation, numerical expressions and *closed-class words* as a precursor of generating the feature set. Vectors representing news texts were created on the basis of a lexical profile of the training set of texts. This lexical profile was determined by two measures: the frequency of a term; and, a weirdness coefficient describing the subject-specificity of a term.

The feature set was created by first selecting the top 5% most frequently occurring words, and from this set, by choosing the words with the highest weirdness coefficient. Subsequently, the 50 most frequent words are selected, excluding spelling mistakes, and numerical expressions and terms too infrequent to provide consistency within a domain are avoided. A high value for the weirdness coefficient is indicative of a word which is uncommon in general language but common in the specialist corpus under examination and is thus a good candidate for a domain term or other word specific to that genre. By disregarding words with a weirdness coefficient lower than a threshold, many *closed-class words* and other terms common in general language are automatically removed. Before we show texts can be categorised using the above method, we digress to briefly outline the Kohonen Self-organising Maps

3.2. Kohonen Self-organising Maps

A SOFM is a neural network and associated learning algorithm that is designed to produce a statistical approximation of the input space by mapping an input in to a two-dimensional output layer (see Kohonen 1997a for an extensive discussion). The approximation is achieved by selection of features that characterise the data, which are output in a topologically ordered map. The Kohonen Self-Organising Map has a close resonance with the *k-means clustering* method, with the additional constraint that cluster centres are located on a regular grid (or some other topographic structure). Furthermore their location on the grid is monotonically related to the pair-wise proximity (Murtagh & Hernández-Pajares, 1995).

The basic SOFM consists of a single layer of neurons formed into a two-dimensional lattice. Each neuron is connected to the input via a set of connections utilising connection weights, just as in a perceptron. There is no ‘output’ of the map, rather the values of each neuron’s weight vector are used to visualise the formed topological ordering. The weight vectors form a cluster prototype that is measured against each input to determine how ‘close’ the vector is to a given cluster. Since the map is two-dimensional and the input typically has a high dimensionality, the SOFM acts as a dimensional squash

allowing the visualisation of features within multi-dimensional data.

Learning is achieved in the SOFM using a competitive algorithm. The Euclidean distance between each training input vector and all weight vectors is determined. The neuron with the weight vector that has the smallest Euclidean distance to the input pattern is termed the winner. To reward the winning neuron its weight vector is adjusted to be ‘closer’ to the input vector, with the amount of adjustment determined by the number of times the training patterns have been presented (via the learning rate). Additionally, all vectors within a defined neighbourhood of the winner are adjusted, essentially forming a cluster of similar values that are seen to be activated by the winner. The neighbourhood size decreases with the number of training cycles, typically using a bubble neighbourhood (a rectangular area) or a Gaussian neighbourhood, both centred on the winning neuron. The adjustment of the weight vector towards the input is achieved by effectively ‘moving’ the weight vector’s direction towards that of the input. This simple process of adjusting ever-smaller neighbourhoods of winners allows the formation of clusters within the lattice. As the number of cycles increases, the clusters become more stable and can be viewed through probing to find winners using test data.

The principal way in which information about the clustering performed by the SOFM learning algorithm is visualised is through probing with a test set to find the winning neurons. The co-location of different winners from different categories highlights the similarity between clusters. The effectiveness of such clusters can be measured by comparing different versions of the map trained on the same data through a technique being developed by Ahmad et al (2001), where Fisher’s Linear Discriminant Rule is used to quantify the discrimination ability of different clusters.

3.3. Limitations of a SOFM

The SOFMs strength lies in its ability to *statistically* summarise the input space. However, it has been shown that the basic SOFM does not always produce a *faithful* approximation (Ritter & Schulten, 1986). This faithful approximation is defined as the proportionality between the density of the weight vectors and the density of the input space. Lin et al (1997) has shown that the SOFM underrepresents high-density regions and overrepresents low-density regions.

3.4. Automatic Categorization of Texts Based on Keywords Using an SOFM

Our text corpus consisted of 100 Associated Press (AP) news wires selected from 10 pre-classified news categories shown in Table 3 together with their icons. The average length of the articles was 622 words.

Text Categories











1	Bioconversion		6	Exportation of Industry	
2	Pollution Recovery		7	Foreign Trade	
3	Alternative Fuels		8	Int. Drug Enforcement	
4	Fossil Fuels		9	Foreign Car Makers	
5	Rain Forests		10	Worldwide Tax Sources	

Table 3: Text categories used in the TIPSTER – SUMMARY program

The 100 AP news wires comprised over 56,000 words. System Quirk was used to compute frequency distribution of words in the AP News wire corpus. The System also has access to the frequency distribution of words in the British National Corpus (Aston and Burnard 1998) a carefully compiled general language corpus. Some of the high weirdness terms, e.g., *drug*, *taxes*, *pollution* and *environmental* are important keywords, but the same cannot be said for ‘terms’ like *billion*, *percent* and *federal*. Usually, proper nouns are also flagged as terms by this method. The feature words identified for the 100 AP News Wire texts are shown in Table 4 according to rank:

1	percent	15	congress	28	dioxide	41	corp
2	tax	16	mexico	29	marine	42	forests
3	billion	17	emissions	30	mazda	43	cocaine
4	drug	18	drugs	31	gases	44	enforcement
5	reagan	19	fuels	32	shale	45	warming
6	cars	20	senate	33	deficit	46	smog
7	taxes	21	auto	34	export	47	ozone
8	environmental	22	proposal	35	recycling	48	Massachusetts
9	pollution	23	gasoline	36	epa	49	imports
10	fuel	24	exports	37	honda	50	automobile
12	federal	25	vehicles	38	methanol	51	trafficking
13	dukakis	26	ohio	39	automakers		
14	bush	27	green-house	40	panama		

Table 4: Feature words identified for the 100 AP News Wire Texts.

Having identified the feature set the training vectors for each of the texts could then be generated. Each vector consisted of binary values indicating the presence or not of each of the feature words determined above.

We have developed a system for creating Kohonen Feature Maps (SANC: Surrey Artificial Network Classifier). The system, after having trained an SOFM, is also capable of testing it. (There are facilities to vary the key parameters associated with the learning algorithm).

The system can be used to test the trained. Furthermore, the system allows the storage of previously trained maps for reference purposes (Ahmad, Vrusias and Ledford 2001).

The results of the Kohonen classifications for full texts are shown in Figure 5. Using symbols to represent each of the locations of the ‘winning node’, the position of each text is indicated across the two-dimensional map (shown in Table 3). It can be seen that the quality of clustering for the full-texts is successful for a range of categories, but especially for categories 9 (FOREIGN CAR MAKERS) and 10

(WORLDWIDE TAX SOURCES). Patterns in categories 1 (BIOCONVERSION), 4 (FOSSIL FUELS), 6 (EXPORTATION OF INDUSTRY) and 8 (INTERNATIONAL DRUG ENFORCEMENT) are also effectively grouped together. The widespread distribution of Class 5 (RAIN FORESTS) shows it to be the worst class on the map.

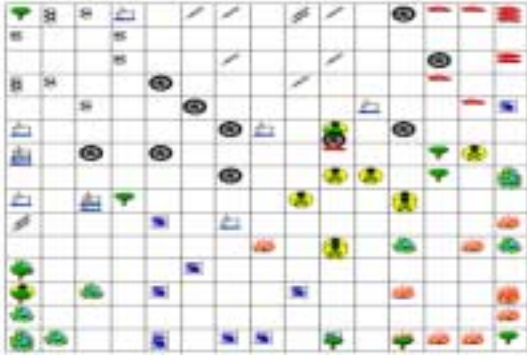


Figure 5: Results of a Full Text Map trained using exponentially decreased neighbourhood and learning rate.

These results for the trained Kohonen map were similar across a number of trials despite variations in training method and learning rate used. Some categories, for example 10 (WORLDWIDE TAX SOURCES), clustered consistently better than others for instance 5 (RAIN FORESTS). By simply counting the number of feature set words that appear in at least nine of the ten texts of each category, the best clustered categories are guaranteed to have some of these words. This reflects the tendency of these categories to cluster well. On the other hand, for a category in the ‘best’ case, only four of the texts share a common feature set word. This difference in classification difficulty was also seen in the TIPSTER results from two human assessors.

3.5. Multiple Categories and Text Categorization

Recall that Reuters News Agency has three categories: “Topic”, “Country” and “Industry”. The total number of different tags, or concepts, defined in these three categories is approximately 1000.

We have created a text corpus of 800 news stories streamed by Reuters in 1997. Each of the news stories is encoded in XML format and has clearly delineated headline, date, writer, text and code fields using XML tagset. The XML-based delineation helps in extracting keywords associated with the Topic, Country and Industry tags. The frequency of each concept was calculated within 800 documents; 80 of the keywords turned out be more frequent than other 920: the distribution of the keywords in the various fields was as follows:

Industry	39	Topic	32	Country	19
-----------------	-----------	--------------	-----------	----------------	-----------

A SOFM was trained for categorising the 102 out of the 800 news stories. The input vector was created from the 80 most frequent keywords associated with the triple, Industry-Topic-Country: the absence and presence of a particular keyword was used to create the input vector for each of the texts. The neural network was trained 100 times. The vector thus created can, in principle, cope with

upto 39 different categories of ‘Industry’, of 32 different ‘Topics’ and ‘19’ different countries. The downside here is that documents comprising references to the 920 keywords may not get classified as well as those that may comprise the 80 categories used in the construction of the input vector.

After the training period, the pre-specified Reuters documents were visualised on the map. As can be seen in Figure 6, the documents associated with each neuron were represented by a blue square. The distribution and the similarity of the documents were based mostly on the “Topic”. On the lower right side of the map, the topics related to “Government/Social” were clustered. The subtopics of “Government/Social”, for example “Sports” and “Art”, were also clustered near this area. The documents categorised as “Management” were found on the lower left corner of the map. “Strategy and Plans”, “Comments/Forecast” and “Economy” follow this as we approach the upper left corner. “European Community” documents were found on the upper right corner of the map.

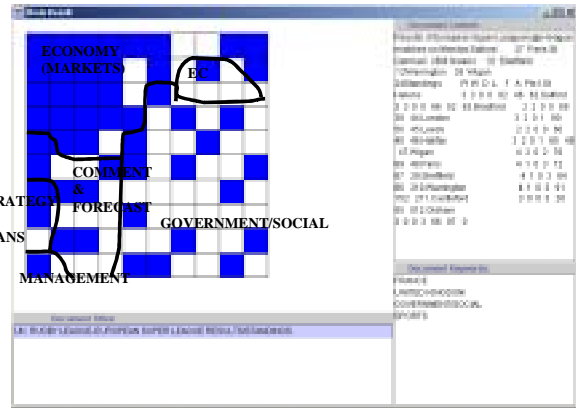


Figure 6: A Categorisation of Reuters news stories using pre-specified category information.

4. Afterword

Our current work involves evaluating the categorisation produced by the method that relies on different distribution of specialist terms in special and general language texts with that of using networks to classify texts that have pre-specified category information as was the case just described.

The pre-specified categories appear to be complex and, as mentioned above subjective in nature. We are currently examining whether a summary of text may give us some indication of the category. The reasoning is as follows: a full news story may contain extraneous material and a good summary will eliminate sentences within the text that are not directly related to the category or categories. Lexical cohesion studies have shown that keywords form the glue that helps to create a cohesive and coherent texts (Hoey 1991). In our previous work on AP news wires (Ahmad, Vrusias and Ledford 2001) we looked at three different types of text streams – headlines only, news summaries and full news items and categorised these texts using self-organising feature maps (SOFM). We found that an SOFM trained on vectors related to summaries

only provides a fairly accurate cluster when compared with vectors related to full text. This work is currently being carried out on the 102 Reuters texts mentioned above.

An analysis shows a vector for the 102 texts using our method based on the weirdness of the keywords within the news stories (Table 5).

Element	Description	Words
1 – 25	<i>Single Words</i> Top 25 simple words with high weirdness and high frequency	inventories, yen analyst directive analysts, merger billion, soccer cents, peso traded, investors, exports allegations quarterly, forecast trading, stocks pesos, shares fiscal, tobacco dealers nickel, earnings
26 – 30	<i>Compound Words:</i> 5 most frequent compound words	shareholder online newsroom chairman worldwide
31 – 40	<i>Proper Nouns</i> 10 proper nouns with high weirdness and high frequency	dorfman ec compuserve kimberly novell chrysler aol saudi microsoft netherlands
41 – 45	<i>Movement Indicators:</i> 5 most frequent downtrend words.	lost risk fall losses falling
46 – 50	<i>Movement Indicators:</i> 5 most frequent up trend words.	up added growth strong high

Table 5: Vector for the 102 Reuters news items (c.1997)

Note that in the above vector we have included movement indicators, proper nouns and compound words together with the single word terms. The 30 keywords and 10 proper nouns/indexical terms, together with 10 movement indicators will help us to define an event. Initial results of this analysis are encouraging in that we obtain the major clusters much like as found in Figure 6

We are currently exploring the notion that news streams will be filtered by using a trained Kohonen SOFM and the filtered text will be used to study market movement. The filter has to be 'cleaned' in that news stories are perishable items with constantly changing subjects – one idea is to re-train the network everyday, towards the end of the day perhaps, with a fixed number of stories which will exclude the very first day of the previous training set and include yesterday's news stories.

Event modelling, especially in noisy and dynamic environments, requires a careful consideration of the key concepts, expressed as keywords, and of indexicals like persons, places, things or ideas which play a crucial role in turning an occurrence, happening or phenomenon into a significant one.

References

- Ahmad, K., Vrusias, L. & Ledford, A. (2001). Choosing Feature Sets for Training and Testing Self-organising Maps: A Case Study. *Neural Computing and Applications*, 10(1), 56-66.
- Ahmad, K. Pragmatics of Specialist Terms and Terminology Management. (1995). In (Ed.) Petra Steffens. *Machine Translation and the Lexicon*. pp. 51-76. Heidelberg: Springer.
- Aston, G. and Burnard, L. *The BNC Handbook: Exploring the British National Corpus with SARA*. 1998. Edinburgh: Edinburgh University Press.
- Gaizauskas, R., Wakao, T., Humphreys, K., Cunningham, H. & Wilks, Y. (1995). Description of the LaSIE system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufmann.
- Hoey, M. (1991). *Patterns of Lexis in Text*. Oxford: Oxford University Press.
- Kaski, S, Honkela, T, Lagus, K & Kohonen T. (1996). Creating an order in digital libraries with self-organising maps. In *Proc. WCNN'96, World Congress on Neural Networks, 1996*, pp 814-817. Lawrence Erlbaum and INNS Press.
- Knowles, F. (1996) *Lexicographical Aspects of Health Metaphors in Financial Texts*. In (Eds.) Martin Gellerstam et al. *Euralex'96 Proceedings (Part II)*. Göteborg, Sweden: Göteborg University. pp 789-796.
- Kohonen, T. (1997a). Exploration of very large databases by self-organizing maps. In *Proceedings of ICNN'97, 1997*, pp. PL1-PL6, IEEE Service Center, Piscataway, NJ.
- Kohonen, T. (1997b). *Self-Organizing Maps*. 2nd Ed. Berlin, Heidelberg, New York: Springer-Verlag.
- Lewis, DD. (1995). Evaluating and optimising autonomous text classification systems. In *SIGIR 95: Proc. of the 18th Annual ACM-SIGIR Conference on Research and Developments in Information Retrieval*. pp 246-254.
- Lin, J.K., Grier, D.G. & Cowan, J.D. (1997). Faithful Representation of Separable Distributions. *Neural Computation*, 9(6), 1305-1320.
- Mani, I. (1998) *The TIPSTER SUMMAC Text Summarization Evaluation*. Mitre Technical Report: MTR 98W0000138, 1998.
- Maybury (1995). *Generating Summaries from Event Data*. *Information Processing and Management*. 31(5), 733-751.
- Murtagh F., Hernández-Pajares M. (1995). The Kohonen Self-Organizing Map Method: An Assessment. *Journal of Classification*, 12, 165-190.
- Rhea, R. (1994). *The Dow Theory*. Burlington: Fraser Publishing Company.
- Ritter, H. & Schulten, K. (1986). On the Stationary State of Kohonen's Self-Organizing Sensory Mapping. *Biological Cybernetics*, 54, 99-106.

Learning IE patterns: a terminology extraction perspective

Roberto Basili, Maria Teresa Pazienza, Fabio Massimo Zanzotto

University of Rome Tor Vergata,
Department of Computer Science, Systems and Production,
00133 Roma (Italy),
{basili, pazienza, zanzotto}@info.uniroma2.it

Abstract

The large-scale applicability of knowledge-based information access systems such as the ones based on Information Extraction techniques strongly depends on the possibility of automatically acquiring the large amount of knowledge required. However, the basic assumption of the IE paradigm, i.e. that the information need is known in advance, limits inherently its applicability since the resulting IE pattern learning algorithms are not generally conceived for the analysis of large corpora if not driven by a specific information need. Since in the terminological studies the corpora and not the information needs already drive the extraction of the knowledge, they offer many insights and mechanisms to automatically model the knowledge content of a coherent text collection. In this paper, we will present a terminological perspective to the acquisition of IE patterns based on a novel algorithm for estimating the domain relevance of the relations among domain concepts. The algorithm and the representation space will be presented. Before starting the discussion, however, we will describe the overall process of building a domain ontology out from an extensional domain model (i.e. the collected domain corpus). Finally, the results of the application of the algorithm over a large domain corpus will be presented and the resulting ontology is discussed.

1. Introduction

The large-scale applicability of knowledge-based information access systems such as the ones based on Information Extraction techniques strongly depends on the possibility of automatically acquiring the large amount of knowledge required. The applicability of these systems over large heterogeneous text collections (e.g. the World Wide Web) may be one of the keys of success of "emerging" information access paradigm such as the Question Answering (QA) and the Automatic Summarisation (AS). In fact, the major strength of the Information Retrieval engines (typically based on the "poor" abstraction of stem) is due more to their wide applicability than to their actual retrieval performances.

A very well assessed approach to Information Access is the paradigm of Information Extraction (MUC-7, 1997; Pazienza, 1997). This latter gave the fertile area where a number of techniques for the automatic acquisition of knowledge have been proposed. However, these learning approaches are focused on the extraction of knowledge needed for the satisfaction of a particular information need (i.e. the one expressed by the template) as the IE paradigm imposes. Therefore, the resulting learning approaches are biased by the fact that they can rely on two important hypothesis limiting their search space. From the one side, the target knowledge domain is generally small and, from the other side, the target information need is very narrow (such as missile launch event in one of the MUC conference). Therefore, the size of the resulting ontology can be kept controlled and the scope of the learning algorithms is a controlled (and small) corpus. In fact, in unsupervised learning techniques as in (Yangarber, 2001; Riloff and Jones, 1999), texts are firstly classified according to their relevance with respect to the particular information need and then particular surface forms somehow related are extracted and retained. The first step narrows the corpus that is given to the second.

However, the basic assumption, i.e. that the information need is known in advance, limits the applicability of the IE paradigm and of the resulting IE pattern learning algorithms. In fact, these latter are not generally conceived for the analysis of large corpora if not driven by a specific information need. If the goal to be achieved is the applicability in large, a different approach has to be undertaken. In such a perspective, the final information needs can not drive the learning phase that should totally rely on the corpus that has to be the source of this information, i.e. it is the final source of information that should suggest the information needs that can be satisfied. This is the typical case a information access system has to face when exposed to an uncontrolled information scenario (e.g. the Web).

Since in the terminological studies the corpus is already the major source of knowledge, they offer many insights and mechanisms to automatically model the knowledge content of a coherent text collection. Here, in fact, the corpus plays the central role of extensional model for the target domain where a domain ontology (i.e. a thesaurus) is extracted from. In this latter, terms and relations among them are generally described. The "operational" notion of *term*, i.e. that the term is the surface representation of a domain concept, allows to define two different levels of analysis: the notion of *admissible surface forms* and the notion of *domain relevance*. The target is generally the extraction of concepts conveyed by nominal phrases and the investigated relations are IS-A and PART-OF. Nevertheless this terminological perspective to the extraction of IE patterns can be adopted for widening the applicability. IE patterns may be considered as domain relations among specific concepts, i.e. typical concepts of the domain and named entity classes that hold by definition the special status of domain concepts.

In this paper, we will present a novel algorithm for estimating the domain relevance of the relations among domain concepts. As for the term, the application of a terminolog-

ical approach to the problem of the discovering the domain relations among concept has to establish:

- which are the surface representations of the target relations;
- which is the estimator of the "domain importance" for the discovered relations.

The algorithm and the representation space will be presented in Sec. 4.. Before starting the discussion, however, we will describe the overall process of building a domain ontology out from a extensional domain model (i.e. the collected domain corpus) in Sec. 2. Finally, the results of the application of the algorithm over a large domain corpus will be presented and the resulting ontology discussed (Sec. 5.).

2. Building an ontology for a large-scale IE system

A large-scale IE system for a news agency should be able to scan news streams. The activity of building the needed knowledge base is therefore a huge task. However, in our opinion, this may be undertaken using some insight given by the terminology extraction practice. News streams are, in fact, coupled with a news classification scheme that can be more or less complex (cf. IPTC standards (IPTC,)), This rough or fine-grained classification over the news items allows the definition of coherent knowledge areas over which terminology extraction techniques can be helpful. Each collection of news items belonging to a class is in fact the extensional model for the underlying domain according to the classifiers.

The process of the knowledge modelling is sketched in the following. Given the corpus as model for the knowledge domain (or class) under investigation, the activities that have to be carried out for building the domain ontology are the following:

1. the definition of the named entity classes
2. a first analysis of the corpus for the acquisition of the most important concepts and relations among the concepts
3. the analysis of the extracted domain knowledge for the definition of the top "event" classes
4. the extraction of all the important concepts and relations among the concepts and their clustering under the defined event classes

For the activities 2 to 4, terminology extraction practice may be very useful with the notions of *admissible surface forms* and of *domain relevance*. The latter is a key notion that helps in showing to the ontology builder only the most relevant IE patterns (a combination of the domain concepts and domain relations). These patterns sorted according the domain relevance estimated by the importance function can drive the definition of the top event classes. The event classes elsewhere referred as "template types" will represent the knowledge the final IE system is able to

make explicit over the particular domain. Finally, since IE patterns are ranked according to their importance, in the activity of clustering this guarantees that the most important events (and generally the most frequent) may be captured by the resulting IE system.

The attention on the clustering activity is somehow one of the major difference between the construction of a domain ontology for an IE system and the one of a terminological knowledge base (TKB) (or thesaurus). This is mainly because of the nature of the typical target knowledge domains. Terminology extraction is mainly conceived for giving a systematic representation of scientific or technological knowledge domains where certain terms are stable and a relatively small number of surface forms are used to convey a domain concept. On the other hand, in the news streams (the areas in which IE system has to find the information) domain concepts and, more often, domain relations are generally conveyed by more than one surface form. It is the equivalence between different event prototypes, i.e. prototypes that specifies the possible instances of the "Who? Where? What? When? Why?" events, that may make the difference.

3. Domain relations among concepts as event prototypes

Event prototypes (or IE patterns) used by IE systems to perform the activity of extracting information are very similar to what a domain relation among domain concepts may look like. Given for instance the financial domain, the prototype necessary to extract a "sell event" from the following news items:

Example 1 *Financial news excerpts*

- (a) *Eon, the German utility formed by the merger of Veba and Viag, is poised to sell its electronics arm to an Anglo-American consortium for about \$2.3bn.*
- (b) *It is understood to be near a deal to sell the Longview smelter for \$150m to McCook Metals.*

may have the following form:

Example 2 *Sell event prototype*

```
sell( (agent:companyNE),  
      (patient:object),  
      (to:companyNE),  
      (for:currencyNE) )
```

i.e. a company typically sells something to a company for a certain amount of money (currencyNE). Here, the two named entity categories, companyNE and currencyNE, are typical concepts of the financial domain and the showed event prototype is a typical domain relation among these concepts.

Due to the difference on the perspective and on the application domain, some adjustments of the techniques developed in terminology extraction are mandatory in the IE pattern extraction problem. As suggested in the example, in IE, a major role is played by named entities. They are not important as surface forms but as generalised forms (i.e. their category). This is a major difference with the general terminology extraction where named entities are important as instances. For instance, *Newton's law* and *Zipf's law* convey very different meaning and are relevant as such and not in a generalised form `personNE's law`. The adoption of TE techniques on the IE tasks requires that named entity categories are considered as typical concepts of the domain. Admissible surface forms also consider the possibility of selecting forms with named entities (e.g. `companyNE_share` where `companyNE` is a named entity category that may be used for detecting *IBM shares* in target text).

Furthermore, in the IE perspective, the definition and the extraction of the domain relations plays a major role. Such a problem is generally neglected in the TE studies because major efforts are spent in the definition of algorithm for extracting and using catalogues for the general relations among terms such as IS-A or PART-OF (Morin, 1999; CON, 1998). The resulting methods are not suitable for the extraction of domain relations.

In order to adopt an TE perspective to the IE pattern learning these two issues have to be faced. In the following section we will present our approach to the extraction of domain relations over large collection of texts.

4. Learning domain relations from large textual collections

The approach to the extraction of domain relations should be completely corpus driven since information needs are not stated in advance. Therefore, given the corpus C , all the relations have to be analysed in order to detect the more important ones. Since the corpus should suggest the typical domain relations in the first phase of the construction of the domain model (cf. Sec. 2.), the target relations should then not to be too far from the admissible surface form as happens for the concept spotting in TE. As for the concept detection, we should then define the admissible surface forms and a function for estimating the domain importance of the given form. However, a minimal abstraction is needed to take into account the relatively free order of the participants when they appear in the actual text as in the above example (Ex. 1). In the following section (Sec. 4.1.), the admissible surface forms and their equivalence are stated and the size of the problem is estimated. On the other hand, an efficient algorithm for the estimation of the importance function based on the frequency of the relations in the target corpus is presented in Sec. 4.2.

4.1. Admissible surface forms: the size of the problem

A relation $r = (rv, (ra_1, ra_2, \dots, ra_n))$ (as the one of the Ex. 2) may be represented in a number of different surface forms. Due to the fact that the corpus should suggest the important relations, we will only consider the realisation of r in verbal phrases. The corpus C is then seen as

a collection of verb contexts $c = (v, (a_1, a_2, \dots, a_n))$ where v is the governing verb and each argument a_i is a couple (g_i, c_i) representing its grammatical role g_i (e.g. subject, object, pp(for), pp(to), etc.) and the concept c_i semantically governing it. A context $c \in C$ is a positive example of the target relation $r \in R$ if $rv = v$ and r partially cover c , i.e. the arguments of r should then appear in any order in the context c .

Given the domain corpus C represented as a collection of verb contexts, the objective is to evaluate the relevance of each possible relation $(r, (ra_1, ra_2, \dots, ra_n))$. The first problem is to estimate how many different relations have to be analysed. This may be obtained after partitioning the corpus C according to the verb governing the contexts. For each verb v , a subset of the corpus is then defined as:

$$C(v) = \{(a_1, \dots, a_n) | (v, (a_1, \dots, a_n)) \in C\} \quad (3)$$

Notice that the notion of context that we use is open to two different 'views'. A lexicalized notion of context is obtained by relying on the full definition. A context $c = (v, ((g_1, c_1), (g_2, c_2), \dots, (g_n, c_n)))$ expresses the governing verb v with the lexical (c_i) and its syntactic role (g_i) for each argument found within a given corpus fragment. c_i is usually a partially generalized surface form. c_i denote thus partially generalized surface forms like `companyNE` (for fragments like *IBM*, *Financial Times*, *Apple Ltd.*) or `companyNE_shares` for structures like *IBM's shares*. If we neglect this rich *lexical* information, and make use a generic concept (e.g. `object`) for the arguments, the remaining information is purely syntactic, making explicit only the grammatical role in the context:

$$c = (v, ((g_1, object), (g_2, object), \dots, (g_n, object)))$$

As a result the following two sets of arguments in contexts of $C(v)$ remain defined:

$$A_\Lambda(v) = \{a | \exists (a_1, \dots, a_n) \in C(v) \wedge \exists i. a_i = a\} \quad (4)$$

$$A_\Sigma(v) = \left\{ \begin{array}{l} (s, object) | \exists i. g_i = s \wedge \\ \exists ((g_1, c_1), \dots, (g_n, c_n)) \in C(v) \end{array} \right\} \quad (5)$$

Given the above sets, $A_\Lambda(v)$ and $A_\Sigma(v)$, the set $R(v)$ of the possible relations for a given v is the following:

$$R(v) = \bigcup_{i=1 \dots MC(v)} R_i(v) \quad (6)$$

where $R_i(v)$ are the collection of individual combinations of exactly i arguments in the set $A(v) = A_\Lambda(v) \cup A_\Sigma(v)$ that are syntactically meaningful. The distinction between lexicalised and syntactic arguments is useful to take into account the fact that some relations may have a recurrent syntactic argument whose filler concept is not recurrent.

If $R(v)$ is the set of all the relations for the investigated verb v , the domain importance of each $r(v) \in R(v)$ should be assessed. Therefore, at least the evaluation of the frequency of the relation $r(v)$ over the corpus $C(v)$ has to be used.

Given the defined sets, the size of the $R(v)$ set is, in the worst case, the following:

$$|R(v)| = \sum_{i=1 \dots MC(v)} \binom{|A(v)| + i - 1}{i} \quad (7)$$

where $MC(v)$ is the maximum context size for the verb v in $C(v)$. It is worth noticing that $|R(v)|$ values lie in a very large range, due to the size of $A(v)$. In the next section we concentrate on a measure of relevance (for the target domain) that allows to systematically reduce the size of the space where pattern selection is applied for each verb v .

4.2. Estimating the importance: Counting efficiently instances of event prototypes

Given the corpus C , the space of the possible relations is huge. This inherent complexity is the result of tackling the argument order freedom that is neglected in (Yangarber, 2001). In order to tackle with the problem, an informed exploration strategy may be settled. This strategy can not take advantage on the biasing given by the awareness of the final information need that is typical of the IE pattern extraction algorithm. However, some observations may be useful for the purpose:

- the target of the analysis is to emphasize the more important relations arising from the domain corpus
- the frequency of a specific relation strictly depends on the frequency of a more general relation

A very simple but effective domain relevance estimator is represented by the frequency of the relation in the corpus. In this perspective, the more important relations are the more frequent. Therefore, the above considerations may reduce the complexity of the search algorithm if only promising relation are explored, i.e. patterns whose generalisations are over a frequency threshold.

The idea is then to drive the analysis using the pattern generalisation that may be obtained projecting the patterns on their "syntactic" counterpart. The projection $\widehat{\Sigma}(r)$ of the relation r over the syntactic space Σ is defined as follows:

$$\widehat{\Sigma}(r) = (\widehat{\Sigma}(ra_1), \dots, \widehat{\Sigma}(ra_m))$$

where $\widehat{\Sigma}(ra_i) = ra_i$ if ra_i is a "syntactic" argument ($ra_i \in A_{\Sigma}(v)$) or $\widehat{\Sigma}(ra_i) = (s_i, object)$ if $ra_i = (g_i, c_i)$ is a lexicalised argument ($ra_i \in A_{\Lambda}(v)$). The resulting search space $R_{\Sigma}(v) = \{\widehat{\Sigma}(r) | r \in R(v)\}$ is greatly smaller than $R(v)$ since $|A_{\Lambda}(v)| \gg |A_{\Sigma}(v)| = \#positions + 2$. This search space can be used for the extraction of the more promising generalised relations. This subset \overline{R}_{Σ} can be used for narrowing the search space of the following step. In fact, when the acceptance threshold is settled, the resultant admissible relations are confined in the following set:

$$\overline{R}(v) \hat{=} \{r | \Sigma(r) \in \overline{R}_{\Sigma}(v)\} \quad (8)$$

¹Notice that, in syntactically meaningful contexts, arguments may appear with multiplicity higher than 1, so that the factorial expression is a useful approximation.

The overall domain importance estimation procedure may take also advantage from the fact that the order of the relation arguments may be fixed after the analysis of the promising syntactic patterns. The final counting activity can be thus performed with a simple sorting algorithm with the $O(k \log(n))$ complexity. In this case n is directly related to the number of context samples in the corpus $C(v)$. The procedure is sketched in the following:

procedure SelectAndRankRelations($R(v), C(v)$)

begin

Select $\overline{R}_{\Sigma}(v) = \{r \in R_{\Sigma}(v) | hits(r, C(v)) \geq K\}$;

Set $L = \emptyset$;

for each $r \in \overline{R}_{\Sigma}(v)$

$L := L \cup prj(C(v), r)$;

$RankR(v) := CountEquals(L)$;

return $RankR(v)$;

end

where $hits(r, C(v))$ is the number of instances of the relation r in $C(v)$ e $prj(C(v), r)$ is the projection of the contexts in $C(v)$ on the syntactic relation r . The procedure *CountEquals(L)* using a standard sorting algorithm counts the repetition of each element in L . Finally, $RankR(v)$ is the set of couples (f, r) where f the frequency of the relation $r \in \overline{R}(v)$ on the corpus.

5. A case study: IE patterns for the financial domain

The above methodology has been applied for the definition of an ontology for a financial domain. The ontology construction steps have been followed. Firstly, an homogeneous collection of texts has been prepared as the model for the target domain, namely a collection of 13,000 news stories of the *Financial Time* over a period of time ranging from 2000 to 2001. The corpus will be hereafter called *FinTimeNews*. The analysis of the corpus has been carried out with the Chaos robust parser (Basili et al., 2000).

In the tables 1 and 2, excerpts of the lists related to the complex concepts and the relations governed by the verb *to make* are respectively shown. The lists are sorted according to their frequency in the *FinTimeNews* corpus (f in the tables). A manual assessed domain relevance is then reported (DR in the tables). The rate of the complex concepts retained as useful exceeds the 60% in the presented top 50 positions. It is worth noticing that many of the complex concepts that have not been judged important for the domain are in fact relevant time indicator. These are not useful for understanding the nature of the domain knowledge but they are precious in the perspective of a IE system for the characterisation of the time stamp of the event. Some of these expression such as *first_half* are in any case typical of the financial jargon, in particular they are used in the declaration of the companies' economic performance.

In the case of the relations governed by the verb *make*, the number of domain relevant relations in the top 50 is around 28%. The other presented relations are generally phraseological use of the same verb.

The sorted lists allows the definition of the top level hierarchy of the possible events in the financial domain.

<i>f</i>	Surface form	<i>DR</i>
2924	last_year	
1739	chief_executive	✓
1138	last_week	
1086	next_year	
956	percentNE_stake	✓
946	entityNE_share	✓
834	last_month	
737	oil_price	
687	joint_venture	✓
641	first_half	
631	pre-tax_profit	✓
618	interest_rate	✓
583	entityNE_yesterday	
575	entityNE_company	✓
551	stake_in_entityNE	✓
499	prime_minister	✓
453	first_time	
438	entityNE_market	✓
431	entityNE_index	✓
429	earnings_per_share	✓
413	share_in_entityNE	✓
412	mobile_phone	
396	profit_of_currencyNE	✓
374	next_month	
361	second_quarter	
358	entityNE_official	
348	second_half	
341	few_year	
341	same_time	
337	entityNE_government	✓
332	next_week	
318	last_night	
316	percentNE_rise	✓
316	end_of_the_year	
309	end_of_dateNE	
299	entityNE_s_share	✓
291	economic_growth	✓
285	recent_year	
281	loss_of_currencyNE	✓
281	central_bank	✓
275	entityNE_deal	✓
269	percentNE_increase	✓
267	percentNE_stake_in_entityNE	✓
248	public_offering	✓
240	executive_of_entityNE	✓
237	net_profit	✓
234	past_year	
234	entityNE_economy	✓
230	acquisition_of_entityNE	✓
229	entityNE_shareholder	✓

Table 1: Complex concepts in *FinTimesNews*

<i>f</i>	Surface form	<i>DR</i>
150	<i>(make,[(diobj,sense)])</i>	
132	<i>(make,[(diobj,money)])</i>	✓
121	<i>(make,[(diobj,profit)])</i>	✓
118	<i>(make,[(diobj,decision)])</i>	
108	<i>(make,[(for,entityNE)])</i>	
106	<i>(make,[(diobj,sense),(subj,null)])</i>	
102	<i>(make,[(in,locationNE)])</i>	
100	<i>(make,[(to,entityNE)])</i>	
100	<i>(make,[(diobj,null),(for,entityNE)])</i>	
95	<i>(make,[(subj,company)])</i>	✓
87	<i>(make,[(diobj,acquisition)])</i>	✓
83	<i>(make,[(for,null),(subj,entityNE)])</i>	
81	<i>(make,[(diobj,null),(to,entityNE)])</i>	
80	<i>(make,[(diobj,null),(in,locationNE)])</i>	
79	<i>(make,[(diobj,progress)])</i>	✓
76	<i>(make,[(in,entityNE)])</i>	
75	<i>(make,[(diobj,null),(subj,company)])</i>	✓
71	<i>(make,[(subj,locationNE)])</i>	
71	<i>(make,[(diobj,use)])</i>	
71	<i>(make,[(diobj,difference)])</i>	
66	<i>(make,[(diobj,use),(of,null)])</i>	
65	<i>(make,[(subj,entityNE),(to,null)])</i>	
60	<i>(make,[(diobj,offer)])</i>	✓
57	<i>(make,[(subj,null),(to,entityNE)])</i>	
57	<i>(make,[(diobj,null),(in,entityNE)])</i>	
55	<i>(make,[(diobj,profit),(subj,null)])</i>	✓
55	<i>(make,[(diobj,null),(subj,locationNE)])</i>	
54	<i>(make,[(diobj,effort)])</i>	
53	<i>(make,[(in,locationNE),(subj,null)])</i>	
53	<i>(make,[(diobj,currencyNE)])</i>	✓
51	<i>(make,[(diobj,mistake)])</i>	
50	<i>(make,[(diobj,null),(subj,entityNE),(to,null)])</i>	
49	<i>(make,[(diobj,debat)])</i>	✓
48	<i>(make,[(for,entityNE),(subj,null)])</i>	
48	<i>(make,[(diobj,money),(subj,null)])</i>	✓
48	<i>(make,[(diobj,bid)])</i>	✓
47	<i>(make,[(diobj,locationNE)])</i>	
46	<i>(make,[(on,null),(subj,entityNE)])</i>	
45	<i>(make,[(diobj,null),(for,entityNE),(subj,null)])</i>	
45	<i>(make,[(diobj,entityNE),(diobj2,null),(subj,null)])</i>	
45	<i>(make,[(diobj,difference),(subj,null)])</i>	
44	<i>(make,[(diobj,sense),(subj,ii)])</i>	
42	<i>(make,[(diobj,progress),(subj,null)])</i>	
42	<i>(make,[(diobj,decision),(subj,null)])</i>	
41	<i>(make,[(diobj,investment)])</i>	✓
40	<i>(make,[(diobj,payment)])</i>	✓
39	<i>(make,[(diobj,case)])</i>	
38	<i>(make,[(diobj2,currencyNE)])</i>	
37	<i>(make,[(diobj,contribution)])</i>	
35	<i>(make,[(with,entityNE)])</i>	
35	<i>(make,[(diobj,loss)])</i>	✓

Table 2: Relations governed by the verb *to make* in *FinTimesNews*

These have been defined as follows:

1. Relationships among companies
 - (a) Acquisition/Selling
 - (b) Cooperation/Splitting
2. Industrial Activities
 - (a) Funding/Capital
 - (b) Company Assets (Financial Performances, Balance Sheet Analysis)
 - (c) Staff Movement (e.g Management Succession)
 - (d) External Communications
3. Company Positioning
 - (a) Position vs. the competitors
 - (b) Market Sector
 - (c) Market Strategies
4. Governmental Activities
 - (a) Tax Reduction/Increase
 - (b) Anti-trust Control
5. Job Market - Mass Employment/Unemployment
6. Stock Market
 - (a) Share Trends
 - (b) Currencies Trends

Once the definition of the top level events has been completed, the discovered event prototypes have been manually clustered according to their class. To give the flavour of the information contained in the produced knowledge base, in the following an excerpt of the event prototypes of the *Company Assets* class are presented:

Company Assets Event Prototypes

(cut,[(subj,entityNE),(diobj,cost)])
(rise,[(subj,profit),(to,currencyNE)])
(rise,[(from,currencyNE),(subj,profit),(to,currencyNE)])
(issue,[(subj,entityNE),(diobj,profit_warning)])
(suffer,[(subj,entityNE),(diobj,loss)])
(report,[(subj,entityNE),(diobj,loss_of_currencyNE)])
(announce,[(subj,entityNE),(diobj,loss_of_currencyNE)])

The analysis of 1,100 patterns give rise to 229 patterns retained as useful for the definition of the event prototypes in one of the give class.

6. Conclusions and future work

In this paper we presented a terminological perspective to the extraction of IE patterns. This corpus driven method is more suitable for a wide application of IE-based systems with respect to learning methods driven by the specific information need. The presented method helps in performing the activities required for building a domain ontology since the concepts and the relations are presented according to their relevance for the target domain.

Many issues are still open and are objective of further research. First of all, a more complete evaluation of the method should be performed with respect to the task of

event recognition. The acquired ontology should be evaluated in order to understand if the level of detail of the event prototypes is deep enough for the experts to classify the event prototypes in the correct class. Therefore, we intend to study the possibility of automatically cluster the event prototypes once the domain top level hierarchy has been defined. We will try here to adopt a booting algorithm and we will study the size of the necessary booting data. Finally, domain relations (i.e. IE patterns) not headed by verbs may be an interesting area of research.

7. References

- Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. 2000. Customizable modular lexicalized parsing. In *Proc. of the 6th International Workshop on Parsing Technology, IWPT2000*, Trento, Italy.
1998. In Didier Bourigault, Christian Jacquemin, and Marie-Claude L'Homme, editors, *Proceedings of the First Workshop on Computational Terminology COMPUTERM'98, held jointly with COLING-ACL'98*, Montreal, Quebec, Canada.
- IPTC. Iptc standards. In *www.iptc.org*.
- Emmanuel Morin. 1999. *Extraction de liens sémantiques entre termes à partir de corpus de textes techniques*. Ph.D. thesis, Université de Nantes, Faculté des Sciences et de Techniques.
- MUC-7. 1997. Proceedings of the seventh message understanding conference(muc-7). In *Columbia, MD*. Morgan Kaufmann.
- Maria Teresa Pazienza. 1997. *Information Extraction. A Multidisciplinary Approach to an Emerging Information Technology*. Number 1299 in LNAI. Springer-Verlag, Heidelberg, Germany.
- Ellen Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*.
- Roman Yangarber. 2001. *Scenario Customization for Information Extraction*. Ph.D. thesis, Courant Institute of Mathematical Sciences, New York University.

Unsupervised Event Clustering in Multilingual News Streams

Martijn Spitters, Wessel Kraaij

Department of Multimedia Technology & Statistics
TNO TPD
P. O. Box 155, 2600 AD Delft
The Netherlands
{spitters, kraaij}@tpd.tno.nl

Abstract

The Topic Detection and Tracking (TDT) benchmark evaluation project embraces a variety of technical challenges for information retrieval research. The TDT topic detection task is concerned with the unsupervised grouping of news stories according to the events they discuss. A detection system must both discover new events as the incoming stories are processed and associate incoming stories with the story clusters created so far. The TNO topic detection system is based on a language modeling approach. The system has been evaluated on a multilingual corpus of approximately 80.000 stories from multiple news sources. For the grouping of stories we combined a simple single pass method to establish an initial clustering and a reallocation method to stabilize the clusters within a certain allowed deferral period. The similarity of an incoming story S_n to an existing cluster C is defined as the average of the similarities of S_n to each story $S_i \in C$. These individual similarities are computed by taking the sum of the generative probabilities $P(S_n|S_i)$ and $P(S_i|S_n)$ where S_i and S_n are modeled as unigram language models. Because these story language models are based on extremely sparse statistics, the word probabilities are smoothed using a background model.

1. Introduction

This paper describes the design and development of a system for the unsupervised grouping of news stories according to the events they discuss. The system has been evaluated on an augmented version of the TDT3 corpus which contains approximately 80.000 stories from multiple news sources, including both text and speech. These sources are newswires, radio and television broadcasts, and internet sites. The source languages are English and Mandarin. The TDT3 corpus is annotated for 120 events, each of which spans both English and Mandarin sources.

The TNO topic detection system is based on a language modeling approach. We had good experience with the application of language models for different IR-related tasks, like ad hoc, cross language, web and spoken document retrieval (Hiemstra and Kraaij, 1999; Kraaij et al., 2000; Hiemstra et al., 2001; Kraaij et al., 2002), filtering (Ekkelenkamp et al., 1999), and multi-document summarization (Kraaij et al., 2001). We also successfully applied language models for topic tracking (Spitters and Kraaij, 2001). However, due to the substantially higher computational complexity of topic detection, it was not trivial to convert our tracking approach into a detection algorithm. In the topic tracking task, events are to be followed individually. Each target event is defined by a small set of training stories that discuss it. Our tracking system estimates a single unigram language model based on the union of these on-topic stories and computes for each incoming story the likelihood according to this topic model. The computational complexity of this process is linear to the input. However, the topic detection task is a highly dynamic process. The topic models are constructed on the fly from the incoming stories. Each incoming story is added to a cluster, and thus changes the corresponding topic model. Experiments showed that reclustering the already processed stories (within the allowed deferral window) is important for a good performance. Reclustering is a computationally

demanding process, since every change in cluster membership lists is reflected in changes in the cluster models, which form the basis for the similarity computation. Therefore we have chosen for a clustering approach which is independent of the (global) cluster models and instead is based on the similarities between individual stories. The advantage of this approach is that the inter-story similarities can be cached, resulting in a significant speed-up of the clustering process.

The remainder of this paper is organized as follows. To familiarize the reader with the TDT framework, section 2 elaborates on the TDT corpora, the TDT research tasks, and the TDT evaluation method. In section 3 we describe in detail our language model-based approach to topic detection. This section also contains a short study into the influence of two different smoothing methods for language models on the detection performance of our system. In section 4 we try to draw some conclusions.

2. The TDT benchmark test

The topic detection and tracking (TDT) benchmark evaluation project¹ was initiated by DARPA in 1996. After a pilot study in 1997, TDT has continued with annual evaluations conducted by the National Institute of Standards and Technology (NIST). Main purpose of the TDT project is to advance the state-of-the-art in determining the topical structure of multilingual news streams from various sources. See (Wayne, 2000) for a detailed overview of the TDT project.

2.1. TDT corpora

Currently, the Linguistic Data Consortium (LDC) has three corpora available to support TDT research² (Cieri et al., 2000). The TDT-Pilot corpus contains newswire and

¹<http://www.nist.gov/speech/tests/tdt>

²<http://www ldc.upenn.edu/Projects/TDT>

transcripts of news broadcasts, all in English, and is annotated for 25 news events. The TDT2 and TDT3 corpora are multilingual (Chinese and English) and contain both audio and text. ASR transcriptions and close captions of the audio data as well as Systran translations of the Chinese data are also provided. TDT2 and TDT3 are completely annotated for 100 and 120 events respectively. Currently, LDC is developing a new TDT corpus (TDT4) which will include Arabic news.

In the TDT evaluation, there are three alternative choices for the form of the audio sources to be processed, namely manual transcriptions, ASR transcriptions, or the sampled audio signal. Three story boundary conditions are supported: reference story boundaries (manually determined correct boundaries), automatic story boundaries (automatically determined errorful boundaries), or no story boundaries (the system must provide its own boundaries). Sites that participate in one of the TDT tasks are required to perform at least one evaluation under shared conditions. See (Doddington and Fiscus, 2001) for the TDT evaluation details.

2.2. TDT research tasks

The TDT benchmark evaluation project embraces a variety of technical challenges for information retrieval research. The goal of *story segmentation* is to segment a stream of data into homogeneous regions, discussing certain events. Given a small number of stories that discuss a certain event, a *tracking* system has the task to detect which stories in the data stream are related to this event and which are not. In *topic detection* there is no knowledge of the events to be detected. A detection system must both discover new events as the incoming stories are processed and associate incoming stories with the event-based story clusters created so far. A task which is very similar to topic detection is *first-story detection*. The goal of this task is to detect, in a chronologically ordered stream of stories, the first story that discusses a certain event. Finally, in *link detection*, the question to be answered is whether or not two stories discuss the same event.

2.3. TDT evaluation method

Topic detection systems are evaluated in terms of their ability to cluster together stories that discuss the same event (or events and activities that are directly connected to the cluster’s seminal event). Detection performance is characterized in terms of the probability of miss and false alarm errors (P_{Miss} and P_{FA}). To speak in terms of the more established and well-known precision and recall measures: a low P_{Miss} corresponds to high recall, while a low P_{FA} corresponds to high precision.

These two error probabilities are combined into a single detection cost C_{Det} , by assigning costs to miss and false alarm errors (Doddington and Fiscus, 2001):

$$C_{Det} = C_{Miss} \cdot P_{Miss} \cdot P_{target} + C_{FA} \cdot P_{FA} \cdot P_{-target} \quad (1)$$

where C_{Miss} and C_{FA} are the costs of a miss and a false alarm respectively; P_{Miss} and P_{FA} are the condi-

tional probabilities of a miss and a false alarm respectively; P_{target} and $P_{-target}$ are the a priori target probabilities ($P_{-target} = 1 - P_{target}$).

Then C_{Det} is normalized to:

$$(C_{Det})_{Norm} = \frac{C_{Det}}{\min(C_{Miss} \cdot P_{target}, C_{FA} \cdot P_{-target})} \quad (2)$$

Detection error probability is estimated by accumulating errors separately for each topic and by taking the average of the error probabilities over topics, with equal weight assigned to each topic. A set of predefined topics is automatically mapped to the system output topics by choosing for each reference topic the system output topic which produces the lowest evaluation cost.

3. Design of a probabilistic topic detection system

This section describes in detail the design of the TNO topic detection system. 3.1. describes our clustering approach. We combined a simple single pass method to establish an initial clustering and a reallocation method to stabilize the clusters within a certain allowed deferral period. In 3.2. we describe our story-cluster similarity measure. An incoming story is compared to an existing cluster by averaging the similarities of the new story S_n to each story in the cluster S_i . These individual similarities are defined as the sum of the generative probabilities $P(S_n|S_i)$ and $P(S_i|S_n)$ where S_i and S_n are modeled as unigram language models. Because these story language models are based on extremely sparse statistics, the word probabilities are smoothed using a background model. Section 3.3. reports on our experiments concerning the application of two different smoothing methods for language models and some contrastive tests with automatic versus manually determined story boundaries.

3.1. Clustering method

Our clustering procedure combines a simple single pass method and a reallocation method. Because the clusters formed by the single pass method are dependent of the order in which the stories are processed, they are merely used to initiate reallocation clustering. However, because in the TDT evaluation a topic detection system may defer its assignment of stories until a limited amount of subsequent source data (10 source files) is processed, the reallocation is restricted to the stories within that deferral period. More specifically, our clustering process involves the following steps:

1. For each new story within the deferral window, compute its similarity to each cluster the system has created so far. There are two options for a story:
 - (a) if the similarity of the story to the closest cluster exceeds a certain threshold, assign the story to that cluster
 - (b) else create a new cluster with the concerning story as its seed

2. When the end of the deferral window is reached, loop through the window stories again and compare each story to each existing cluster. There are three options for a story:

- (a) a story may switch to another cluster if the similarity to that cluster exceeds both the similarity to its current cluster and the threshold
- (b) if neither the similarity to its current cluster nor the similarity to any other cluster exceeds the threshold, create a new cluster with the concerning story as its seed
- (c) if the similarity to its current cluster exceeds the threshold as well as the similarities to all other clusters, the story stays in its current cluster

Step 2 is repeated until all clusters are stable, that is, when 2c is true for each story.

The combination of a cluster initialization step and a re-allocation step has previously (successfully) been used for topic detection by a.o. BBN (Walls et al., 1999) and Dragon (Yamron et al., 2000).

The reclustering step is important for a good performance of the detection system. However, the fact that every change in a cluster membership list means that the cluster language model would have to be reestimated, makes it a computationally demanding process. Therefore we have chosen for an approach which does not use the global cluster language models (contrary to our topic tracking approach) but instead is based on the similarities between individual stories. The similarity of an incoming story S_n to an existing cluster C is defined as the average of the similarities of S_n to each story $S_i \in C$. The advantage of this approach is that the inter-story similarities can be cached, resulting in a significant acceleration of the clustering process. These inter-story similarities are computed using a two-way language modeling approach, which is discussed in detail in the following section.

A cluster which has not changed for an uninterrupted period of fifteen days is frozen, which means that it is no longer considered an ‘active event’. The cluster is removed from the list of candidate clusters for new stories. This cluster evolution monitoring has two advantages. First of all it limits the computational complexity, because the number of clusters a story has to be compared with stays within certain bounds. Second, it can be argued that restricting the temporal extent of an event is beneficial for detection performance because it prevents different events with similar vocabulary (like different attacks or political elections) to be grouped together (Yang et al., 1999).

3.2. Language model-based similarity

The basic idea behind the language modeling approach to information retrieval is to estimate a (usually unigram) language model for each document and to rank documents by the probability that the document model generated the query. Absolute probabilities are not important for ranking in the IR situation. For other applications, i.e. topic tracking and also topic detection, scores have to be comparable

on an absolute scale. For tracking, we found that modeling similarity as a likelihood ratio and normalizing this likelihood ratio by the (test) story length was adequate (Spitters and Kraaij, 2001). This normalized likelihood ratio is presented in equation (3), where $LLR_{Norm}(T_1, T_2, \dots, T_n | S_k)$ denotes the normalized log likelihood ratio of a story consisting of the terms T_1, \dots, T_n given the story S_k in comparison with background model \mathcal{B} .

$$LLR_{Norm}(T_1, T_2, \dots, T_n | S_k) = \frac{1}{n} \log \sum_{i=1}^n \frac{P(T_i | S_k)}{P(T_i | \mathcal{B})} \quad (3)$$

In our clustering approach, the similarity between two stories S_n and S_i is based on a combination of the probability that the language model representing S_n generated story S_i and the reverse: the probability that the language model representing S_i generated story S_n . This approach results in the symmetrical similarity measure, presented in the following equation:

$$Sim(S_n, S_i) = LLR_{Norm}(S_n | S_i) + LLR_{Norm}(S_i | S_n) \quad (4)$$

Because the language models are estimated based on very limited amounts of text (single stories), it is very important that the word probabilities are smoothed using some background model. We performed a short study into the influence of two different smoothing methods on the performance of our detection system: Bayesian smoothing using Dirichlet priors and Jelinek-Mercer smoothing. The details of these smoothing methods and the results of our experiments are described in the following section.

3.3. Smoothing

Recent experiments at CMU have shown that different smoothing methods have different characteristics (Zhai and Lafferty, 2001a). For title ad hoc queries, Zhai and Lafferty found Dirichlet smoothing to be more effective than linear interpolation (Jelinek-Mercer smoothing). Both methods start from the idea that the probability estimate for unseen terms: $P_u(T_i | S_k)$ is modeled as a coefficient α_s times the background collection based estimate: $P_u(T_i | S_k) = \alpha_s \cdot P(T_i | \mathcal{B})$. A crucial difference between Dirichlet and Jelinek-Mercer smoothing is that the smoothing coefficient is dependent on the story length for Dirichlet, reflecting the fact that probability estimates are more reliable for longer stories. Formula (5) shows the weighting formula for Dirichlet smoothing, where $c(T_i | S_k)$ is the term frequency of term T_i in story S_k , $\sum_w c(T_i; S_k)$ is the length of story S_k and μ is a constant. The smoothing coefficient α_s is in this case $\frac{\mu}{\sum_w c(T_i; S_k) + \mu}$, whereas the smoothing coefficient is λ in the Jelinek-Mercer based model (formula (6)).

$$P(T_1, T_2, \dots, T_n | S_k) = \prod_{i=1}^n \frac{c(T_i; S_k) + \mu P(T_i | \mathcal{B})}{\sum_w c(T_i; S_k) + \mu} \quad (5)$$

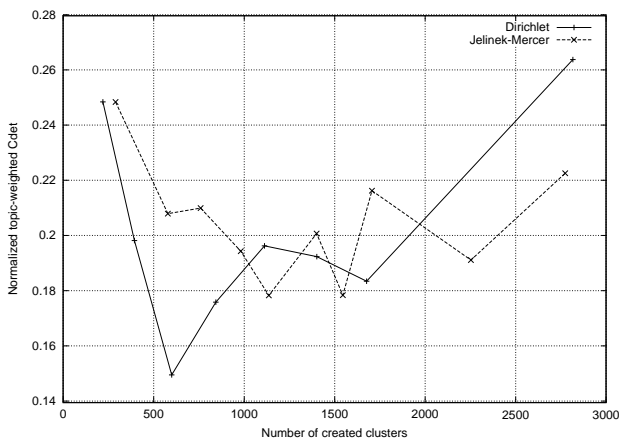


Figure 1: C_{Det} at different decision thresholds for two smoothing methods (Dirichlet and Jelinek-Mercer), performed on the TDT2 stories from April 1998, using automatic boundaries.

$$P(T_1, T_2, \dots, T_n | S_k) = \prod_{i=1}^n \lambda P(T_i | \mathcal{B}) + (1 - \lambda) P(T_i | S_k) \quad (6)$$

For our official TDT2001 detection run, we applied Dirichlet smoothing with $\mu = 2000$. Our hypothesis was that Dirichlet smoothing would lead to improved performance, since story lengths vary considerably in the TDT corpus, and Dirichlet performed better than Jelinek-Mercer smoothing on a small test corpus (one month of stories from the TDT2 corpus) using the automatic story boundaries and ASR transcriptions of the audio (the primary topic detection evaluation requires these conditions). The results of this experiment are plotted in Figure (1).

We performed some post-hoc experiments on this same test set using reference story boundaries instead of automatic story boundaries and were surprised to find that Jelinek-Mercer performed better than Dirichlet under that condition, even when we varied μ (see equation (5)). Figure (2) shows the results. It is too early to draw conclusions from these experiments, since the test set was small and we did not explore the complete parameter space. However, one explanation could be the observation from Zhai and Lafferty (Zhai and Lafferty, 2001b; Zhai and Lafferty, 2001a) that smoothing has two functions: i) improving the maximum likelihood estimates ii) generate common words in the query. The latter function is especially important for longer queries since they contain more common words.

In the topic detection task we use language models to generate stories instead of queries. Since stories are considerably longer than TREC title queries, it is probably important that the smoothed model generates common words with proper “idf”-like probabilities. The TREC experiments show that the two roles of smoothing have an inverse interaction with the query length. Dirichlet is a good strategy for the first smoothing role (avoiding the assignment of a zero probability to an unseen word) while Jelinek-Mercer is better for the second role (weighting query terms in an

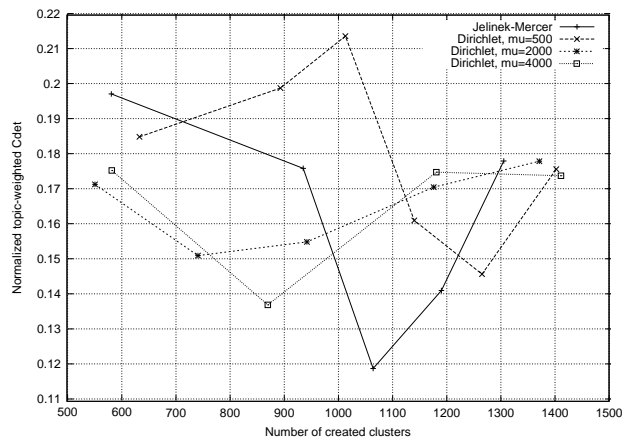


Figure 2: C_{Det} at different decision thresholds for two smoothing methods (Dirichlet and Jelinek-Mercer), performed on the TDT2 stories from April 1998, using reference boundaries.

idf-like fashion) (Zhai and Lafferty, 2001a). The longer the “queries” are, the more important the second function will become. This phenomenon might be an explanation for the fact that Dirichlet performs best under the automatic story boundary condition, and Jelinek-Mercer under the reference story boundary condition, since the former has shorter stories than the latter (median: 62 versus 114). Further experiments are needed, including a validation of a combined Dirichlet/Jelinek-Mercer smoothing scheme for the TDT tasks.

4. Conclusions and future work

We think that the choice to use normalized likelihood ratios as the basis of a similarity measure was the key for the good performance of our system. Like in the tracking task, a proper normalized similarity measure is of utmost importance. Simply adding the generative probabilities $P(S_n | S_i)$ and $P(S_i | S_n)$ proved to work well to “symmetrize” the similarity measure. The accuracy of a language model-based clustering approach which is independent of the (global) cluster models and instead is based on the similarities between individual stories surpassed our expectations. However, we intend to check whether a similarity measure based on the global cluster model would enhance the results. The results of some initial post-hoc experiments indicate that the Jelinek-Mercer smoothing method works better than Dirichlet smoothing for manually segmented data, while the Dirichlet method yields better performance than Jelinek-Mercer on automatically segmented data. Further investigation is necessary to draw definite conclusions.

5. References

- C. Cieri, D. Graff, M. Liberman, N. Martey, and S. Strassel. 2000. Large multilingual broadcast news corpora for cooperative research in topic detection and tracking: The TDT2 and TDT3 corpus efforts. *Proceedings of the Language Resources and Evaluation Conference (LREC2000)*.

- G. Doddington and J. Fiscus. 2001. The year 2001 topic detection and tracking (TDT2001) task definition and evaluation plan. Technical Report v. 1.0, National Institute of Standards and Technology.
- R. Ekkelenkamp, W. Kraaij, and D. van Leeuwen. 1999. TNO TREC-7 site report: SDR and filtering. *Proceedings of the Seventh Text Retrieval Conference (TREC-7)*, pages 519–526.
- D. Hiemstra and W. Kraaij. 1999. Twenty-one at trec-7: Ad hoc and cross language track. *Proceedings of the Seventh Text Retrieval Conference (TREC-7)*, pages 227–238.
- D. Hiemstra, W. Kraaij, R. Pohlmann, and T. Westerveld. 2001. Twenty-one at CLEF 2000: Translation resources, merging strategies and relevance feedback. *Proceedings of the CLEF 2000 Cross-Language Text Retrieval System Evaluation Campaign*.
- W. Kraaij, R. Pohlmann, and D. Hiemstra. 2000. Twenty-one at TREC-8: using language technology for information retrieval. *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, pages 282–299.
- W. Kraaij, M. Spitters, and M. van der Heijden. 2001. Combining a mixture language model and naive bayes for multi-document summarisation. *Notebook papers of the Document Understanding Conference (DUC 2001)*.
- W. Kraaij, T. Westerveld, and D. Hiemstra. 2002. The importance of prior probabilities for entry page search. *Proceedings of the 25th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, To Appear.
- M. Spitters and W. Kraaij. 2001. Using language models for tracking events of interest over time. *Proceedings of the Workshop on Language Models for Information Retrieval (LMIR 2001)*, pages 60–65.
- F. Walls, H. Jin, S. Sista, and P. van Mulbregt. 1999. Topic detection in broadcast news. *Proceedings of the DARPA Broadcast News Workshop*.
- C.L. Wayne. 2000. Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. *Proceedings of the Language Resources and Evaluation Conference (LREC2000)*, pages 1487–1494.
- J.P. Yamron, S. Knecht, and P. van Mulbregt. 2000. Dragon’s tracking and detection system for the TDT2000 evaluation. *Notebook papers of the Topic Detection and Tracking Workshop (TDT) 2000*.
- Y. Yang, J. Carbonell, R. Brown, T. Pierce, B.T. Archibald, and X. Liu. 1999. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems: Special Issue on Applications of Intelligent Information Retrieval*, 14(4):32–43.
- C. Zhai and J. Lafferty. 2001a. Dual role of smoothing in the language modeling approach. *Proceedings of the Workshop on Language Models for Information Retrieval (LMIR) 2001*, pages 31–36.
- C. Zhai and J. Lafferty. 2001b. A study of smoothing methods for language models applied to ad hoc information retrieval. *Proceedings of SIGIR 2001*.