Australasian Language Technology Association Workshop 2009

Proceedings of the Workshop

Editors:
Luiz Augusto Pizzato
Rolf Schwitter

3-4 December 2009
University of New South Wales
Sydney, Australia

Australasian Language Technology Association Workshop 2009
(ALTA 2009)
URL: `http://www.alta.asn.au/events/alta2009/index.html`

Sponsors:

Volume 7, 2009
ISSN: 1834-7037

Preface

This volume contains the papers accepted for presentation at the Australasian Language Technology Workshop (ALTA) 2009, held as part of HCSNet (ARC Network in Human Communication Science) SummerFest at the University of New South Wales, Sydney, Australia on December 3-4, 2009. This is the seventh annual installment of the ALTA workshop in its most-recent incarnation, and the continuation of an annual workshop series that has existed in various forms Down Under since the early 1990s.

The goals of the workshop are:

- to bring together the growing Language Technology (LT) community in Australia and New Zealand and encourage interactions;
- to encourage interactions and collaboration within this community and with the wider international LT community;
- to foster interaction between academic and industrial researchers, to encourage dissemination of research results;
- to provide a forum for the discussion of new and ongoing research and projects;
- to provide an opportunity for the broader artificial intelligence community to become aware of local LT research; and, finally, to increase visibility of LT research in Australia, New Zealand and overseas.

This years ALTA Workshop includes full paper presentations as well as short paper presentations. Of the 27 papers submitted to ALTA, 18 papers were selected by the program committee for publication. One of these 18 papers was withdrawn by the authors after the review process. The remaining 17 papers appear in these proceedings; 10 papers as full papers (9 pages in length) and 7 papers as short papers (5 pages in length).

Each paper was independently peer-reviewed by at least two members of an international program committee, in accordance with the DEST requirements for E1 conference publications. The ALTA Workshop is classified as a Tier B conference in the Computing Research and Education Association of Australasia (CORE) classification system which will form the basis of ranking computer science conference publications for the ARC.

We would like to thank all the authors who submitted papers to ALTA; the members of the program committee for the time and effort they put into the review process; and to our international plenary speaker, Prof. David Traum, University of Southern California, Institute for Creative Technologies.

Our thanks also go to HCSNet for its invaluable financial and organisational support, and for bringing together a large number of research communities under one multidisciplinary roof. In particular, we would like to thank Kym Buckley, Chris Cassidy, and Ben Phelan for their perfect organization of this event.

Luiz Augusto Pizzato and Rolf Schwitter
Program Co-Chairs

ALTA 2009 Committees

Workshop Co-Chairs

- Luiz Pizzato(University of Sydney)
- Rolf Schwitter (Macquarie University)

Workshop Local Organiser

- HCSNet (The ARC Network in Human Communication Science)

Program Committee

- Timothy Baldwin (University of Melbourne)
- Steven Bird (University of Melbourne)
- Lawrence Cavedon (National ICT Australia, Melbourne)
- Eric Choi (National ICT Australia, Sydney)
- Peter Clark (Boeing)
- Nathalie Colineau (CSIRO)
- Nigel Collier (NII – National Institute of Informatics, Japan)
- Robert Dale (Macquarie University)
- Jean-Yves Delort (CMCRC)
- Mark Dras (Macquarie University)
- Dominique Estival (University of Sydney)
- Caroline Gasperin (USP/ICMC, Brazil)
- Ben Hachey (CMCRC)
- Michael Hess (University of Zuerich)
- Jerry R. Hobbs (USC/ISI)
- Achim Hoffmann (University of New South Wales)
- Matthew Honnibal (University of Sydney)
- Alistair Knott (University of Otago)
- Kazunori Komatani (Kyoto University)
- Andrew Lampert (CSIRO)
- David Martinez (University of Melbourne)
- Tara E. McIntosh (University of Sydney)
- Diego Moll Aliod (Macquarie University)
- Eamonn Newman (Dublin City University)
- Cecile Paris (CSIRO)
- David M. W. Powers (Flinders University)
- Adam Saulwick (DSTO)
- Harold Somers (Dublin City University)
- Jette Viethen (Macquarie University)
- Martin Volk (University of Zuerich)
- Menno van Zaanen (Tilburg University)
- Simon Zwarts (Macquarie University)

ALTA 2009 Program

Thursday, 3rd December 2009

9:00-10:00	HCSNet Plenary talk: Eckart Altenmuller
10:00-10:30	Coffee break
10:30-10:45	ALTA Opening remarks
Session 1 - 10:45 - 12:30	
	Full papers
10:45-11:15	Ben Hachey *Evaluation of Generic Relation Identification*
11:15-11:45	Will Radford, Ben Hachey, James R Curran and Maria Milosavljevic *Tracking Information Flow in Financial Text*
	Short papers
11:45-12:00	Christopher Chua, Maria Milosavljevic and James R. Curran *A Sentiment Detection Engine for Internet Stock Message Boards*
12:00-12:15	Su Nam Kim, Timothy Baldwin and Min-Yen Kan *Extracting Domain-Specific Words - A Statistical Approach*
12:15-12:30	Jon Patrick and Min Li *A Cascade Approach to Extracting Medication Events*
12:30-2:00	Lunch
2:00-3:00	HCSNet Plenary talk: Jonathan Harrington
3:00-3:30	Coffee break
Session 2 - 3:30-4:30	
	Full papers
3:30-4:00	Nicky Ringland, Joel Nothman, Tara Murphy and James R Curran *Classifying articles in English and German Wikipedia*
	Short papers
4:00-4:15	Sam Tardif, James R Curran and Tara Murphy *Improved Text Categorisation for Wikipedia Named Entities*
4:15-4:30	Aleksandar Igic, Catherine Watson, Jonathan Teutenberg, Elizabeth Broadbent, Rie Tamagawa and Bruce MacDonald *Towards a flexible platform for voice accent and expression selection on a Healthcare Robot*
4:30-5:30	**2009 Annual General Meeting of the Australasian Language Technology Association**

Friday, 4th December 2009

9:00-10:00	HCSNet Plenary talk: David Traum
10:00-10:30	Coffee break
Session 3 - 10:30-12:30	
	Full papers
10:30-11:00	Simon Musgrave and Michael Haugh *The AusNC Project: Plans, Progress and Implications for Language Technology*
11:00-11:30	James Breen and Timothy Baldwin *Corpus-based Extraction of Japanese Compound Verbs*
11:30-12:00	Meladel Mistica, Wayan Arka, Timothy Baldwin and Avery Andrews *Double Double, Morphology and Trouble: Looking into Reduplication in Indonesian*
12:00-12:30	Sze-Meng Jojo Wong and Mark Dras *Contrastive Analysis and Native Language Identification*
12:30-2:00	Lunch
2:00-3:00	HCSNet Plenary talk: Mark Sanderson
3:00-3:30	Coffee break
Session 4 - 3:30-5:30	
	Full papers
3:30-4:00	Jonathan K. Kummerfeld, Jessika Roesner and James Curran *Faster parsing and supertagging model estimation*
4:00-4:30	Tim Dawborn and James R. Curran *CCG parsing with one syntactic structure per n-gram*
4:30-5:00	Colin White and Rolf Schwitter *An Update on PENG Light*
	Short papers
5:00-5:15	James Constable and James Curran *Integrating Verb-Particle Constructions into CCG Parsing*
5:15-5:30	Bahadorreza Ofoghi and John Yearwood *From Lexical Entailment to Recognizing Textual Entailment Using Linguistic Resources*
5:30	Close

Contents

HCSNet Plenary Speaker:

Spoken Dialogue Models for Virtual Humans

David Traum
Institute for Creative Technologies, Marina del Rey
University of Southern California, USA
traum@ict.usc.edu

Abstract

In this talk, I will survey several different kinds of dialogue models in use at the University of Southern California's Institute for Creative Technologies. These models differ in complexity, robustness, ease of authoring, and thus suitability for different kinds of projects, ranging from research prototypes to systems in use for training applications or as presentation tools accessible to the general public. The models include a text-classification approach in which answers are selected from an authored set and no semantic reasoning is performed, "traditional" form-filling dialogue, a merging of the previous two approaches along with finite state networks for local dialogue structure, and a more advanced information-state model that is closely linked with AI planners and emotion reasoners.

Generic Relation Identification: Models and Evaluation

Ben Hachey
Centre for Language Technology
Macquarie University
NSW 2109 Australia

Capital Markets CRC Limited
GPO Box 970
Sydney NSW 2001

bhachey@cmcrc.com

Abstract

Generic relation identification (GRI) aims to build models of relation-forming entity pairs that can be transferred across domains without modification of model parameters. GRI has high utility in terms of cheap components for applications like summarisation, automated data exploration and initialisation of bootstrapping of relation extraction. A detailed evaluation of GRI is presented for the first time, including explicit tests of portability between newswire and biomedical domains. Experimental results show that a novel approach incorporating dependency parsing is better in terms of recall. And, accuracy is shown to be comparable across domains.

[^{place} American] saxophonist [^{person} David Murray] recruited [^{person} Amidu Berry] and DJ [^{person} Awadi] from [^{organisation} PBS].

Figure 1: Example input to GRI task (from ACE 2004). Square brackets indicate the extent of entity mentions with type as italicised superscript.

Entity 1	Entity 2	Relation Type
American	David Murray	CITIZEN/RESIDENT
David Murray	Amidu Berry	BUSINESS
David Murray	Awadi	BUSINESS
Amidu Berry	PBS	MEMBER-OF-GROUP
Awadi	PBS	MEMBER-OF-GROUP

Table 1: Example output from GRI task. Relation types are not part of the relation identification task but are given here for purposes of illustration.

1 Introduction

Relation extraction (RE) aims to identify mentions of relations in text. A relation mention is defined as a predicate ranging over two arguments, where an argument represents concepts, objects or people in the world and the predicate describes the type of stative association or interaction that holds between the things represented by the arguments. Input to the RE task consists of source documents with entity mention markup (e.g., Figure 1). The output is a list of relation-forming entity mention pairs and a label indicating the type of relation that holds between them (e.g., Table 1). This paper addresses the relation identification task, which identifies pairs of relation-forming entity mentions (e.g., "David Murray" and "Amidu Berry" in the example).

Relation extraction (RE) can be addressed using supervised (Zelenko et al., 2005; Blitzer et al., 2006), bootstrapping (Brin, 1998; Riloff and Jones, 1999; Agichtein and Gravano, 2000; Hassan et al., 2006) or generic approaches (Conrad and Utt, 1994; Hasegawa et al., 2004). One way to characterise these different approaches is in terms of adaptation cost, i.e. the amount of work necessary to adapt them to a new domain or task. In these terms, supervised approaches (including rule engineering and supervised machine learning) incur the highest cost as systems need to be built largely from scratch for each new domain. Bootstrapping approaches incur less cost as they require only a small amount of seed data. And generic approaches provide domain adaptation for free as parameters do not need to be modified for new domains or tasks. Another way to char-

acterise these approaches is in terms of the ontology creation problems they address, i.e. whether they address only the instantiation task where instances are added to an ontology in a new domain given a *relation schema* (the taxonomy of relation types to be identified) or whether they also address the task of learning the relation schema for the new domain. In these terms, supervised approaches and bootstrapping approaches address only the ontology instantiation problem while generic approaches also address the problem of learning relation schemas from data. The trade-off is in terms of accuracy, where generic approaches suffer when compared to supervised and bootstrapping approaches. However, generic approaches have high utility in terms of developing cheap components for applications like paraphrase acquisition (Hasegawa et al., 2005), on-demand information extraction (Sekine, 2006) and automatic summarisation (Hachey, 2009a). And, they could be used for initialisation of semi-supervised bootstrapping of relation extraction.

This paper contains the first detailed evaluation of generic relation identification (GRI), including explicit tests of portability between newswire and biomedical domains. GRI can be split into two sub-tasks, where input consists of source documents with entity mention markup (as in Figure 1). The first sub-task has the goal of identifying relation-forming entity mention pairs and outputs a list of co-occurring entity mention pairs (e.g., Table 1). The second sub-task has the goal of applying a ranking over co-occurring pairs that indicates the strength of association. This ranking might be used for filtering low confidence relations or in weighting schemes for extrinsic applications (e.g., automatic summarisation). The experiments here focus primarily on the identification sub-task, which is evaluated with respect to gold standard data. Experiments are reported that compare window-based models (e.g., setting a threshold on the number of intervening tokens). Results show that a novel approach incorporating intervening words and dependency paths is better in terms of recall while being statistically indistinguishable in terms of precision and f-score. Furthermore, performance is shown to be comparable when porting from news to biomedical text without modification of model parameters.

Author	Co-occur Window	Constraints
Hasegawa	W/in 5 words	NA
Zhang	Sentence	Spanning parse
Conrad	W/in 25, 100 words	NA
Smith	Sentence	NA
Filatova	Sentence	Verbal connector

Table 2: Approaches from the GRI literature.

2 Related Work

Table 2 contains an overview of approaches from the GRI literature. The first column (Author) contains the first author of the approaches referenced in the following text. The first two rows correspond to approaches that address relation identification and characterisation; the third and fourth rows correspond to approaches that focus on the GRI task; and the fifth row corresponds to a related approach to generic *event* identification and characterisation. The second column (Co-occur Window) describes the co-occurrence window for identifying entity mention pairs (e.g., W/in 5 words means that entity mention pairs need to occur within five tokens of each other). The third column (Constraints) describes any additional constraints placed on entity mention pairs.

Hasegawa et al. (2004) introduce the task of relation discovery (using unsupervised techniques to annotate pairs of associated objects with a relation type derived from the textual context). Their work includes a simple approach to GRI where all pairs of entity mentions within 5 tokens of each other are considered to be co-occurring. No motivation is given for choosing 5 as the threshold. In subsequent work, Zhang et al. (2005) incorporate syntactic parsing (Collins, 1999) into their approach to GRI. All pairs of entities in the same sentence are considered to be co-occurring provided that there is a spanning parse. Neither Hasegawa et al. nor Zhang et al. explicitly evaluate their approaches to relation identification.

Filatova and Hatzivassiloglou (2003) describe related work that aims to extract entity pair associations that constitute what they term atomic events. They consider any pair of entity mentions co-occurring within a sentence to be possible participants in event descriptions and they add a constraint requiring that a verbal 'connector' (i.e., a verb or a noun that is a WordNet hy-

ponym of *event* or *activity*) be present in the intervening token context between the entity mentions. The authors present a limited evaluation of their approach to relation identification which suggests reasonable precision. However, it is based on manual analysis of the system output so is not repeatable. Furthermore, it does not address recall and it does not compare the system to any lower or upper bounds on accuracy.

Conrad and Utt (1994) present seminal work on mining pairs of entities from large text collections using statistical measures of association to rank named entity pairs based on co-occurrence. They propose windows of size 25 and 100, which means that any other entity mention within 25 or 100 tokens to the right or left of a given entity mention is considered to co-occur. These window sizes are chosen as they roughly approximate mean sizes of paragraphs and documents in their data. The authors do not specify which window size they use for their evaluation. A manual evaluation of system output is reported, which suggests reasonable performance but is not repeatable.

Smith (2002) considers all pairs of entities in the same sentence to be co-occurring. He performs an evaluation using a corpus of nineteenth century American historical documents. Extracted entity pairs are compared to a curated resource, which contains expert assessments of the severity of battles in the American civil war. Again, this suggests reasonable performance but is not repeatable. Furthermore, Smith (2002) does not compare to lower or upper bounds.

In the literature on supervised relation extraction, e.g. Liu et al. (2007), features based on parse trees have been used successfully. However, beyond requiring a spanning parse tree (Zhang et al., 2005), no previous approaches have investigated the use of syntactic parsing to constrain GRI. The current work investigates the use of domain-neutral co-occurrence windows for GRI that are based on paths connecting entity mention pairs through syntactic parse trees. Furthermore, it presents the first detailed evaluation of GRI on publicly available relation extraction data.

3 Evaluation

To address previous shortcomings, a principled framework is introduced that uses gold standard relation extraction data to optimise and evaluate GRI models. This is derived from news data from the Automatic Content Extraction (ACE) 2004 and 2005 shared tasks[1] and biomedical data derived from the BioInfer corpus.[2] The ACE 2004 data is used for development experiments. The ACE 2005 data serves as the held-out news test set and the BioInfer data serves as the biomedical test set. See Hachey (2009b) for details of the data preparation and experimental setup.

GRI T/F	ACE2004	ACE2005	BioInfer
True	949	558	1591
False	8304	5587	4252
Total	9253	6145	5843

Table 3: Distribution of relations.

Accuracy is measured in terms of precision (P) and recall (R):

$$P = \frac{NumCorrect}{TotalSystemPairs} \qquad R = \frac{NumCorrect}{TotalGoldPairs}$$

And, f-score (F) is calculated in the standard way: $F = 2PR/(P + R)$. Paired Wilcoxon signed ranks tests[3] across entity pair sub-domains are used to check for significant differences between systems. Sub-domains are formed by taking just those relations between two entities of given types (e.g., *Person-Organisation*, *Gene-Protein*). Table 3 contains the count of same-sentence entity mention pairs that constitute relation mentions (True) and those that are not (False). In the ACE 2004 and 2005 data sets, this results respectively in 949 and 558 true relation mentions spread across seven entity pair sub-domains. In the BioInfer data set, this results in 1591 true relation mentions spread across seven entity pair sub-domains.

The evaluation here also introduces an upper bound for GRI based on human agreement. This is calculated by first obtaining a mapping from entity mentions marked by annotators to entity mentions in the adjudicated gold standard annotation. The mapping used here is derived from the ACE 2005 evaluation script, which computes an

[1] `http://www.nist.gov/speech/tests/ace/`

[2] `http://mars.cs.utu.fi/BinInfer`

[3] The paired Wilcoxon signed ranks test is a non-parametric analogue of the paired t test. The null hypothesis is that the two populations from which the scores are sampled are identical. Following convention, the null hypothesis is rejected for values of p less than or equal 0.05.

optimised one-to-one mapping based on maximal character overlap between system and gold standard entity mention strings. Given this mapping, it is possible to determine for each possible entity mention pair whether the annotators marked a relation mention. Interestingly, the annotators have high agreement with the adjudicated data set in terms of precision at 0.906 and lower agreement in terms of recall at 0.675. This suggests that the annotators rarely marked bad relation mentions but each missed a number of relation mentions that the other annotator marked. The mean human f-score agreement is 0.773.

4 Models

The GRI task can be generalised in terms of the GENERICRELATIONID algorithm in Figure 2. This takes as input an array of entity mentions E and the Boolean function ISPAIR. The ISPAIR function returns true if two entity mention indices constitute a co-occurring pair and false otherwise. Figure 2 includes the ISPAIR$_{baseline}$ function as an example, which simply counts all pairs of entity mentions occurring in the same sentence as relation-forming pairs like Smith (2002). The GENERICRELATIONID algorithm starts by initialising the set of entity mention pairs $\mathcal{P}$ to the empty set. It then loops over all possible pairs of entities from E, which is assumed to be sorted in terms of the order of occurrence. Pairs are added to $\mathcal{P}$ if the text describes a relation between them. The experiments here will be based on different definitions of the ISPAIR function, based on intervening token windows and dependency path windows.[4]

Atomic Events The first model of entity mention co-occurrence is based on the approach to identifying atomic events from Filatova and Hatzivassiloglou (2003). This uses an ISPAIR$_{event}$ function that accepts all pairs of entity mentions that 1) occur in the same sentence and 2) have a verbal 'connector' (i.e., a verb or a noun that is a WordNet hyponym of *event* or *activity*) in the intervening context.

[4]Additional experiments not reported here also explored learnt ISPAIR functions using decision trees and various combinations of generic features. However, these models did not generalise across domains.

```
GENERICRELATIONID: E, ISPAIR
1   P ← {}
2   i ← 0
3   while i ≤ length(E)
4         j ← i + 1
5         while j ≤ length(E)
6               if ISPAIR(i, j)
7                     P ← P ∪ [i, j]
8               j ← j + 1
9         i ← i + 1
10  return P
```

```
ISPAIR_baseline : i, j
1   if sent(i) = sent(j)
2         return true
3   else
4         return false
```

Figure 2: Algorithm for generic relation identification with baseline function for identifying co-occurring entity mention pairs.

Intervening Token Windows The next model is based on intervening token windows (Toks). It uses an ISPAIR$_{toks}$ function that counts all pairs of entity mentions that 1) occur in the same sentence and 2) have t or fewer intervening tokens. Most previous GRI work has used some variant of this model. Hasegawa et al. (2004), for example, use the ISPAIR$_{toks}$ function but do not motivate their threshold of t=5.

Figure 3 contains optimisation results for setting the intervening token threshold t on the news development data (ACE 2004). The shaded bars correspond to mean f-scores (actual value printed above the bars) for different settings of t (specified along the bottom of the horizontal axis). The best f-score is shown in bold. Values that are statistically distinguishable from the best ($p \leq 0.05$) are underlined. The results suggest that the best performance is achieved with t set to 2, though this is not reliably different from scores for t=1 and t=4 which suggests a range of optimal values from 1 to 4. For the comparisons in the rest of this paper, the Toks model should be assumed to have t set to 2 unless stated otherwise. Recall (R) and precision (P) are plotted as dotted grey and solid black lines respectively and are closest to being balanced at t=1.

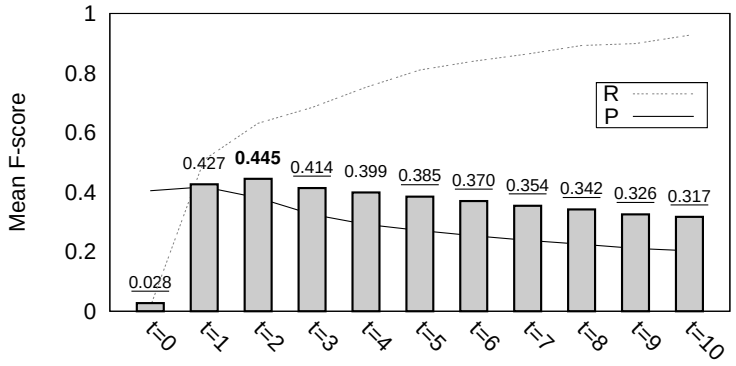

Figure 3: Window size results for token-based model.

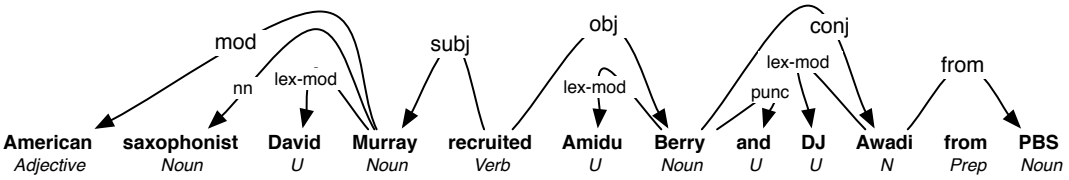

Figure 4: Dependency parse for example sentence.

Dependency Path Windows The experiments here also consider a novel approach to modelling entity mention co-occurrence that is based on syntactic governor-dependency relations (Deps). This uses an IsPair$_{deps}$ function that counts all pairs of entity mentions that 1) occur in the same sentence and 2) have d or fewer intervening token nodes on the shortest dependency path connecting the two entity mentions. Dependency paths are derived using the Minipar software (Lin, 1998), which produces 1) directional links from governors to their dependent lexical items and 2) grammatical relation types (e.g., *subject*, *object*). Figure 4 contains the Minipar parse of the example sentence from Figure 1. The shortest dependency paths between all candidate entity mention pairs are extracted from the parse graph. The path between "American" and "David Murray", for example, consists of a direct *modifier* (mod) relation with zero intervening word token nodes. The path between "David Murray" and "Awadi", on the other hand, passes through one word token node ("recruited") after post-processing operations that pass governor-dependency relations along chains of conjoined tokens, resulting in a *obj* relation between recruited and Awadi.

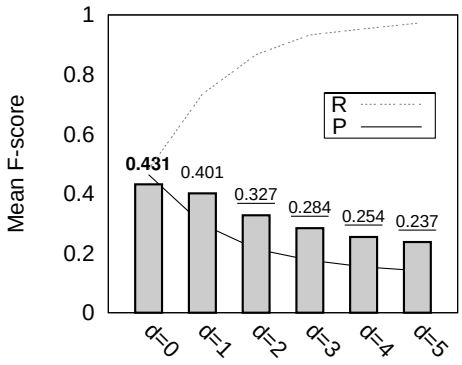

Figure 5: Window size results for dependency-based model.

Figure 5 contains optimisation results for setting the dependency path threshold d on the news development data (ACE 2004). The shaded bars correspond to mean f-score. The best f-score is shown in bold and is achieved at d=0 (which should be assumed from here). Values that are statistically distinguishable are underlined. Results here suggest a range of optimal values from d=0 to d=1. Recall (R) and precision (P) are plotted as dotted grey and solid black lines respectively and are closest to being balanced at d=0.

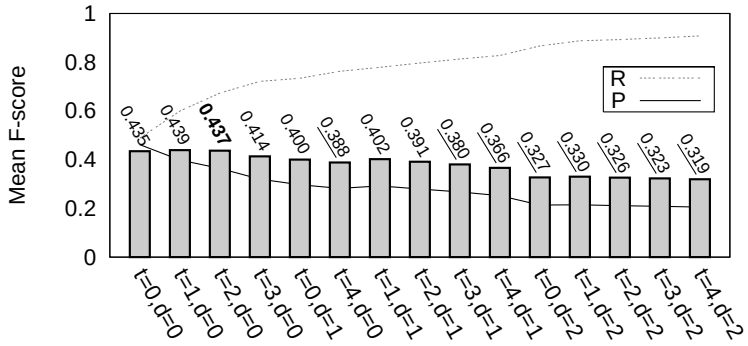

Figure 6: Window size results for combined (token and dependency) model.

Combined Windows Finally, the current work also introduces an entity mention co-occurrence model that combines token and dependency windows (Comb). It uses an IsPair$_{comb}$ function that counts all pairs of entity mentions that 1) occur in the same sentence and 2) either have t or fewer intervening tokens or have d or fewer intervening dependency path nodes. Based on tuning experiments on the news development data (ACE 2004), the thresholds here are set to t=1 and d=0.

Figure 6 contains joint optimisation results for the intervening token (t) and dependency path (d) thresholds on the news development data (ACE 2004). The optimal system is chosen in terms of the mean rank of f-scores across entity pair subdomains. The best mean rank is achieved with t=2 and d=0. Values that are statistically distinguishable from the best are underlined. The results suggest a range of optimal settings with t ranging from 0 to 2 and d ranging from 0 to 1.

5 Results

Table 4 contains P, R and F results. The best score for each measure is in bold and scores that are statistically distinguishable from the best ($p \leq 0.05$) are underlined. The baseline system considers all pairs in the same sentence to be relations.

Which window function is best for identifying relation mentions? The highest f-score on the news test data is obtained using the dependency path model, though this is not statistically distinguishable from the Toks or Comb models. In terms of recall, the Comb model obtains the highest score (0.538), which is significantly better than the Toks and Deps models. The Deps model, however, obtains a precision score that is significantly better than the Comb model. For the current work, the combined model is considered to be the best as it achieves the highest recall while the f-score is statistically indistinguishable from the other models. The prioritisation of recall is motivated by the fact that weighting is generally applied to co-occurring entity pairs for applications of GRI. For example, relation mining approaches from the related literature (Conrad and Utt, 1994; Smith, 2002) use statistical measures of association such as pointwise mutual information, ϕ^2 and log likelihood ratio to estimate association strengths. Thus, a certain amount of noise in GRI should be acceptable if the subsequent weighting scheme is assumed to give higher weight to true relation-forming entity pairs.

How does system performance compare to human performance? The main difference is in terms of precision, where the Comb model performs far worse than the Human upper bound (0.906). However, while Comb recall is significantly worse than Human recall (0.675), the difference is not large. Furthermore, inter-annotator agreement on ACE is a very strong upper bound for the GRI task as the annotators are given detailed guidelines that provide a prescriptive notion of what counts as a relation mention. The GRI task, on the other hand, is not guided by a predefined schema and GRI predicts a number of relation mentions that are incorrect with respect to the gold standard annotation but could arguably be considered true relation mentions.

a) ACE 2005 (News Test Set)

	P	R	F
Baseline	0.110	1.000	0.195
Event	0.050	0.392	0.083
Toks	0.291	0.510	0.342
Deps	**0.456**	0.392	**0.360**
Comb	0.277	**0.538**	0.332
Human	0.906	0.675	0.773

b) BioInfer (Biomedical Test Set)

	P	R	F
Baseline	0.268	1.000	0.415
Event	0.186	0.418	0.247
Toks	**0.527**	0.388	0.422
Deps	0.450	0.302	0.349
Comb	0.500	**0.454**	**0.453**
Human	NA	NA	NA

Table 4: Comparison of P, R and F on news and biomedical test sets. The best score in each column is in bold and those that are statistically distinguishable from the best are underlined.

Does model performance generalise across domains? In the biomedical domain, the Comb model performs best in terms of f-score with a score of 0.453 though it is statistically indistinguishable from the Toks model. This is a stronger result than in the news domain where there was no significant differences among the f-scores of the Toks, Deps and Comb models. Consistent with the news domain, there are no significant differences among the precision scores of the Toks, Deps and Comb models and, importantly, the Comb model is significantly better than the Toks and Deps models in terms of recall in both domains. Interestingly, the f-score of the Baseline model is statistically indistinguishable from the Comb model on the biomedical data. Since Baseline recall is the same for both domains (1.000), this is due to higher precision (0.268 as opposed to 0.110). This suggests that the biomedical GRI task is easier due to the higher proportion of true relation-forming pairs (27% compared to approximately 10% for the ACE data sets). This may be artificially high, however, since the BioInfer creators selectively sampled sentences that include mentions of proteins that are known to interact. The biomedical result is consistent with the news result, however, in that Comb precision is significantly better than Baseline precision on both domains.

6 Discussion

Recall and precision of the Event model The low recall of the Event model with respect to the other models is not surprising due to the constraint requiring an intervening event word. The low precision, however, indicates that the constraint is not helpful as a method to capture long-distance relation mentions based on intervening token windows. The Event model does particularly poorly on the ACE 2005 *GPE-GPE* and BioInfer *Protein-ProteinFamily* entity pair sub-domains due to the fact that true pairs rarely have a verbal connector in the intervening token context. True relation mentions in the ACE 2005 *GPE-GPE* sub-domain tend to be geographical part-of relations where the two entity mentions are adjacent (e.g., the relation between the *GPE* entity mention "Peoria" and the *GPE* entity mention "Illinois" in the fragment "Peoria, Illinois"). And, true relation mentions in the BioInfer *Protein-ProteinFamily* sub-domain tend to be appositives (e.g., the relation between the *Protein* entity mention "cofilin" and the *ProteinFamily* entity mention "actin-binding protein" in the fragment "cofilin, a ubiquitous actin-binding protein") or nominal modifiers (e.g., the relation between the *ProteinFamily* entity mention "cyclin-dependent kinase inhibitors" and the *Protein* entity mention "p57" in the fragment "the cyclin-dependent kinase inhibitors (CKIs) p27 and p57").

Error Analysis For each entity pair sub-domain, ten instances were chosen randomly from the set of erroneously classified instances. These were manually inspected in order to characterise the types of errors made by the combined (Comb) GRI system. This suggests that the majority of false positive errors in both the news and biomedical data sets (81% and 54% respectively) can be considered implicit relation mentions (i.e., the relation is not explicitly stated but

is more or less implicit given the context of the sentence). For example, our system posits a false positive relation between "Gul" and "Erdogan" in the sentence "Unlike the soft-spoken Gul, Erdogan has a reputation as a fighter." These types of false positives are not necessarily problematic in applications of GRE. In fact, these implicit relation mentions are likely to be helpful in applications, e.g. representing the conceptual content of a sentence for extractive summarisation (Hachey, 2009a). One not unexpected difference between domains is that there were considerably more false negative errors in the biomedical data that could be attributed to parsing errors (15% as opposed to 5% in the news data).

Comparison of ranking methods Since it is trivial to improve recall simply by increasing token or dependency thresholds, improvements in f-scores require models with higher precision. One possible approach for improving precision would be to incorporate methods from the literature (Conrad and Utt, 1994; Smith, 2002) for ranking entity mention pairs using statistical measures of association, such as pair probability (Pr), log-likelihood (G^2), ϕ^2, and pointwise mutual information (PMI). Table 5 contains correlation (point-biserial) scores that compare rank weights obtained from these measures with a binary variable indicating whether the instance constitutes a true relation mention according to the annotation. Following Cohen (1988), values over 0.10 (typeset in italicised bold font) are considered to indicate a small effect and values over 0.30 (typeset in bold font) are considered to indicate a medium effect. The table suggests that a threshold filtering low values of PMI would be the best filter for the ACE 2005 test set (small to medium correlation of 0.273, 0.356, 0.168 and 0.326 respectively for the Baseline, Toks, Deps and Comb models). On the BioInfer test set, by contrast, no measure has consistent correlation across systems and effect sizes are largely negligible. The highest correlation is 0.116 for G^2 on the Comb system. While this effect is small, in conjunction with the ACE 2005 results, it suggests that G^2 would be the better ranking method for domain-neutral relation identification.

a) ACE 2005 (News Test Set)

	Pr	G^2	ϕ^2	PMI
Baseline	-0.093	***0.108***	***0.262***	***0.273***
Toks	-0.098	***0.250***	**0.329**	**0.356**
Deps	-0.092	0.067	***0.145***	***0.168***
Comb	-0.091	***0.219***	***0.294***	**0.326**

b) BioInfer (Biomedical Test Set)

	Pr	G^2	ϕ^2	PMI
Baseline	0.030	0.037	***0.105***	0.073
Toks	***0.114***	***0.107***	-0.009	-0.004
Deps	0.056	0.070	-0.023	-0.008
Comb	0.081	***0.116***	0.003	0.041

Table 5: Point-biserial correlation analysis comparing a true relation mention indicator feature to various approaches for ranking GRI predictions by pair association strength.

7 Conclusions

This paper presented a detailed evaluation of the generic relation identification (GRI) task, providing a comparison of various window-based models for the first time. It compared the intervening token window approach (Toks) from the literature to a novel GRI approach based on windows defined over dependency paths (Deps). In addition, it introduced a combined approach (Comb) that integrates the intervening token and dependency path models. Models were optimised on gold standard data in the news domain and applied directly to data from the news and biomedical domains for testing. The use of the ACE 2005 data for a news test set allowed comparison to a human upper bound for the task.

Model comparison suggested that the Deps and Comb models are best. In particular, the Comb approach performed reliably better than the other models in terms of recall while maintaining statistically indistinguishable precision and f-score. High recall models were prioritised here based on the fact that applications of generic relation extraction generally incorporate a mechanism for ranking identified relation mentions. Experiments and analysis suggest that GRI accuracy is comparable when applying the newswire-optimised models directly to the biomedical domain.

Acknowledgments

This work was supported by Scottish Enterprise Edinburgh-Stanford Link grant R37588 as part of the EASIE project at the University of Edinburgh. It would not have been possible without the guidance of Claire Grover and Mirella Lapata. I would also like to thank Robert Gaizauskas and Steve Renals for thoughtful feedback.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, San Antonio, TX, USA.

John Blitzer, Mark Dredze, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *Proceedings of the EDBT International Workshop on the Web and Databases*, Valencia, Spain.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2005. Automatic relation extraction with model order selection and discriminative label identification. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing*, Jeju Island, Korea.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2006. Unsupervised relation disambiguation with order identification capabilities. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia.

Jacob Cohen. 1988. *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum Associates, Inc., San Diego, CA, second edition.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Jack G. Conrad and Mary Hunter Utt. 1994. A system for discovering relationships by feature extraction from text databases. In *Proceedings of the 17th SIGIR*, Melbourne, Australia.

Elena Filatova and Vasileios Hatzivassiloglou. 2003. Marking atomic events in sets of related texts. In Nicolas Nicolov, Kalina Bontcheva, Galia Angelova, and Ruslan Mitkov, editors, *Recent Advances in Natural Language Processing III*. John Benjamins, Amsterdam/Philadelphia.

Ben Hachey. 2009a. Multi-document summarisation using generic relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Singapore.

Ben Hachey. 2009b. *Towards Generic Relation Extraction*. Ph.D. thesis, University of Edinburgh.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd ACL*, Barcelona, Spain.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2005. Unsupervised paraphrase acquisition via relation discovery. Technical report, Proteus Project, Computer Science Department, New York University.

Hany Hassan, Ahmed Hassan, and Sara Noeman. 2006. Graph based semi-supervised approach for information extraction. In *Proceedings of the TextGraphs: The 2nd Workshop on Graph Based Methods for Natural Language Processing*, New York, NY, USA.

Tor-Kristian Jenssen, Astrid Lægreid, Jan Komorowski, and Eivind Hovig. 2001. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28(1):21–28.

Dekang Lin. 1998. Dependency-based evaluation of MINIPAR. In *Proceedings of the LREC Workshop Evaluation of Parsing Systems*, Granada, Spain.

Yudong Liu, Zhongmin Shi, and Anoop Sarkar. 2007. Exploiting rich syntactic information for relationship extraction from biomedical articles. In *Proceedings of NAACL-HLT*, Rochester, NY, USA.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence*, Orlando, FL, USA.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of the COLING/ACL Main Conference Poster Sessions*, Sydney, Australia.

David A. Smith. 2002. Detecting and browsing events in unstructured text. In *Proceedings of the 25th SIGIR*, Tampere, Finland.

Dmitry Zelenko, Chinatsu Aone, and Jason Tibbetts. 2005. Trainable evidence extraction system (TEES). In *Internationl Conference on Intelligence Analysis*, McLean, VA, USA.

Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering relations from a large raw corpus using tree similarity-based clustering. In *Proceedings of the 2nd IJCNLP*, Jeju Island, Korea.

Tracking Information Flow in Financial Text

Will Radford [†‡] **Ben Hachey** [‡◇] **James R. Curran** [†‡] **Maria Milosavljevic** [‡]

School of Information Technologies [†]
University of Sydney
NSW 2006, Australia

Capital Markets CRC [‡]
55 Harrington Street
NSW 2000, Australia

Centre for Language Technology [◇]
Macquarie University
NSW 2109, Australia

{wradford,james}@it.usyd.edu.au {bhachey,maria}@cmcrc.com

Abstract

Information is fundamental to Finance, and understanding how it flows from official sources to news agencies is a central problem. Readers need to digest information rapidly from high volume news feeds, which often contain duplicate and irrelevant stories, to gain a competitive advantage. We propose a text categorisation task over pairs of official announcements and news stories to identify whether the story repeats announcement information and/or adds value. Using features based on the intersection of the texts and relative timing, our system identifies information flow at 89.5% F-score and three types of journalistic contribution at 73.4% to 85.7% F-score. Evaluation against majority annotator decision performs 13% better than a bag-of-words baseline.

1 Introduction

Financial news is an important resource for capital market participants and plays a central role in how they interact with the market. Companies must continuously disclose any information "a reasonable person would expect to have a material effect on the price or value of the entity's securities" (ASX, 2008). News agencies publish Finance stories that report on a broad range of events. Some stories report facts directly from announcements and may add value by presenting background knowledge, expert analysis or editorial commentary. The financial environment rewards participants that are alert and responsive to incoming information (Zaheer and Zaheer, 1997) and automated analysis of information flow is highly advantageous.

We define information flow between a pair of documents as when one document repeats information from the other. Textual similarity is central to this and has been addressed in a variety of research areas. Plagiarism detection concentrates on verbatim duplication of sections of text (Brin et al., 1995; Wise, 1996), while Information Retrieval techniques assess similarity at the broader topic level (Manning et al., 2008). Text Reuse examines a finer-grained notion of similarity (Metzler et al., 2005) between the verbatim copying and topic similarity. Topic detection and tracking (Allan et al., 1998a) focuses on tracking emerging events at a topic level over a news feed.

We examine two sources: the Australian Securities Exchange (ASX)[1] official announcements and the Reuters NewsScope Archive (RNA)[2], both of which release time-stamped documents tagged with one or more company stock *ticker* codes. In this research, ASX-RNA document pairs that share the same ticker and were published within a time window are extracted. The ASX-RNA pairs are annotated to indicate information flow (LINK) and, if so, whether the story is the first to report an announcement (FIRST), background (BACK) or analysis (ANLY) content. We formulate four tasks classifying whether each label applies to ASX-RNA pairs.

We design textual and temporal features to model information flow between the ASX-RNA pairs. The intersection of unigrams and bigrams from their texts and titles provides a baseline approach. Set-theoretic bags-of-words, similarity scores, sentence and number matches, and common sequence counts are used to capture textual similarity. Temporal features such as the pair publication times and lag represent the market news cycle. In the LINK classification experiments using Maximum Entropy models, we achieve 89.5% F-score and between 73.4% and 85.7% F-score in the BACK, FIRST and ANLY experiments. Evaluated against annotator majority decision, the system scores 13 points above a bag-of-words baseline F-score. With this new task and dataset, we demonstrate that it feasible to track information flow in financial markets.

[1] http://asx.com.au

[2] http://thomsonreuters.com/products_services/financial/financial_products/event_driven_trading/newsscope_archive

2 Background

Global news agencies operate on a 24 hour news-cycle in a highly competitive environment and are under pressure to report events as quickly as possible. Apart from timely reporting, they must isolate the salient facts from source material, simplifying them if necessary. Commonly available background information about people or events is provided to place the story in context. As well as reporting existing information, news sources generate novel information in the form of analysis, editorial content and commentary.

Identifying and measuring the value and timeliness of their contribution is a principal goal of our study. Textual similarity is the core of our approach to the information flow problem and has been explored by many research areas. In the information flow context, we propose that textual similarity will model ASX announcement facts and figures reported in RNA stories.

Information Retrieval provides many fundamental techniques for textual similarity. Perhaps the most simplistic of these is *bag-of-words*, which represents a document as an unordered collection of its words that acts as "a quantitative digest" (Manning et al., 2008). Stopword filtering and weighting functions such as TFIDF (Spärck Jones, 1973) attempt to emphasise information-bearing, or unusual, words. Having represented the text in these ways, vector space models treat them as vector parameters to a cosine function to quantify their similarity (Salton et al., 1975). Despite their simple model of language, these methods are robust and effective.

Plagiarism Detection uses concepts of textual similarity to identify wholesale copying of text or source code that indicates academic misconduct (Brin et al., 1995; Wise, 1996). Although the pathological case of verbatim copying is reasonably easy to detect, exact matching methods can be circumvented by simply reordering sections or changing a few words. Fingerprint techniques have separated documents into meaningful chunks, typically sentences, and sequences of these are hashed for later comparison against new documents. This reduces the complexity of the matching operation and allows the system to scale to large numbers of documents.

Fingerprinting techniques are also used in Co-derivative Document Detection which identifies documents that share a common antecedent. Rather than direct copies, co-derivative documents are those where "long blocks of text are largely preserved, but possibly with intermittent modifications, and some original text is added" (Hoad and Zobel, 2003).

Text Reuse explores the reformulation and restatement of short phrases, part of a similarity spectrum between the specific matches of plagiarism detection and IR's topic similarity (Metzler et al., 2005). Motivated by the concentration of research at either end of this spectrum, the authors aim to track text and facts through corpora at sentence granularity using a variety of similarity measures. Clough et al. (2002) use similarity scores as features to classify newspaper stories as *wholly*, *partially* or *not* derived from UK Press Association newswires. They achieve their best *wholly/partially* F-score of 88.2% at the expense of 64.9% *not* F-score using Naïve Bayes classifiers.

Topic Detection and Tracking (TDT) was part of the TREC programme and focussed on events: "something that happens at a particular time and place" (Allan et al., 1998b). Subtasks, including Event Tracking and Link Detection, encouraged a wide range of approaches including relevance models (Lavrenko et al., 2002) and linguistic features such as noun phrase heads, synonyms and verb semantic classes (Hatzivassiloglou et al., 1999).

Novelty Detection was a later TREC task and models "an application where a user is skimming a set of documents, and the system highlights new, on-topic information" (Soboroff, 2004). Rather than TDT's document oriented approach, the input data is a sequence of sentences related to a topic and is a finer-grained task. Interestingly, the notion of novelty is often encoded as text *dissimilarity* with the already topic-related set of preceding sentences.

The Sentence Alignment task uses a loose notion of textual similarity to align sentences and their translations in parallel corpora and is typically a preprocessing step for Machine Translation training. Differing languages rule out word matching and so approaches tend to address structural features. Brown et al. (1991) report good results using sentence word length to align English and French sentences from Canadian Parliamentary Hansards, as do Gale and Church (1991), who use sentence character length.

News stories and announcements are inextricably linked to their release time and modelling temporal features is important. "Information streams" can be modelled as a 'bursty' time-series using a Hidden Markov Model over hidden states that specify an emission rate (Kleinberg, 2003). Highly-ranked bursts tend to reveal emerging technical terms and language change and the "landmark" documents these appear in is analogous to TDT. Gruhl et al. (2004) consider information flow as an epidemic, using hyperlinks and weighting TFIDF as word use changes.

Our approach appropriates some of these textual similarity techniques and, with temporal features, effectively models information flow.

3 Data

The information flow task requires that we collect documents from both primary and secondary sources cov-

Year	ASX	RNA Stories
2003	66,233	1,901,722
2004	80,570	1,954,259
2005	90,484	2,053,525
2006	102,235	2,298,462

Table 1: Document count per year.

Source	Count	Text (%)
ASX	10,404	83.9
RNA	8,277	99.6

Table 2: Document type distribution and text coverage.

ering the same time span and tickers. Sirca[3] provides ASX official announcements and RNA stories to subscribers. Table 1 shows the document count per year for our entire dataset. The overarching trend is that the volume of ASX and RNA data increases with time, though it is worth noting that the count for RNA data includes all Reuters stories released globally, which explains the disparity in size.

Table 2 describes our experimental dataset: a subset of the ASX and RNA datasets, all chosen from an 18 month period from the beginning of 2005. We show the counts of documents in each source and the proportion of those documents which yielded usable text. To filter the RNA stories specific to the ASX market, we select only those marked with ASX tickers and the English language tag. We choose 403 tickers from the ASX200 index[4] over the last day of each year from 2002 to 2008 to identify large and newsworthy companies.

The broad scope of the ASX's continuous disclosure rules means that almost any type of document can appear as an announcement. While these are all in PDF format, the dataset includes short letters, corporations law forms, long annual reports and presentation slides. In addition to these differences in length and form, companies' different industries mean that a wide variety of genres and topics can appear. For any content-level processing, the announcement text must be extracted from the PDF file, which may include scanned or faxed documents. Text for 83.9% of documents was extracted using the PDFBox[5] Java libraries. Plain-text metadata is also included, specifying the publishing timestamp, title and related tickers of each announcement.

The RNA data collects together stories taken from the global Reuters news feed and represents a unique multi-lingual resource. Each story is made up of a sequence of distinct *events* based on the Reuters workflow. Figure 1, taken from the RNA documentation, shows an example of the evolution of a story. A newsworthy event might consist of alerts concisely stating the main information, followed by a newsbreak with a headline, two to four paragraphs and then any number of updates to Reuters' coverage. We use a unique 'story key' found in each event to aggregate them into a story, reconstituting the text to its final, canonical form. In addition to the text and title, each story is tagged with lists of relevant tickers, languages, topics and geographical areas. Only 55.7% of RNA stories are tagged with one ticker, in contrast to the 92.7% of ASX announcements. RNA stories, as such, are more likely to report about more than one company and possibly contain different threads of information.

[3] http://www.sirca.org.au
[4] http://www.standardandpoors.com
[5] http://incubator.apache.org/pdfbox

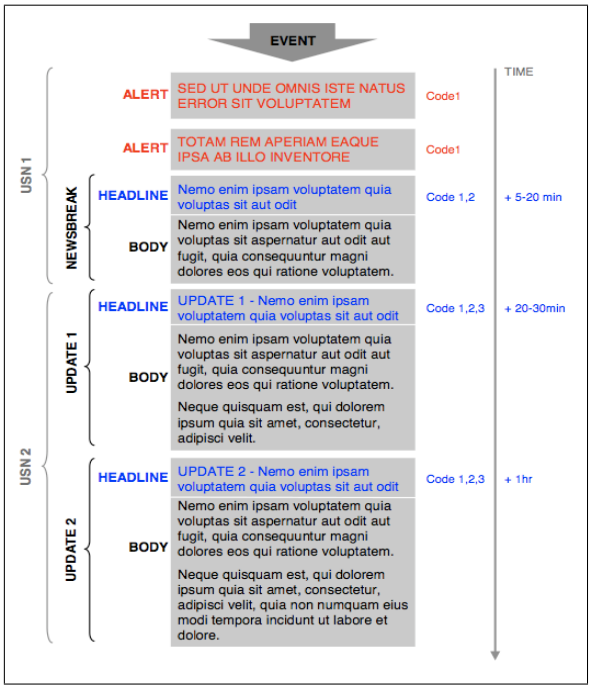

Figure 1: The evolution of an RNA story.

Link	Text
FIRST	...Record BHP profit of $2.45 million...
BACK	...BHP has been moving into NSW...
ANLY	...The profit exceeds expectation, said...

Table 3: Examples of RNA story journalistic contribution given the ASX announcement information: *BHP posted record annual profits of $2.45 million.* .

4 Annotation Scheme

We developed a scheme that codifies information flow in Finance text. The scheme describes two phenomena: whether an RNA story contains information from an ASX announcement (LINK) and, if so, the journal-

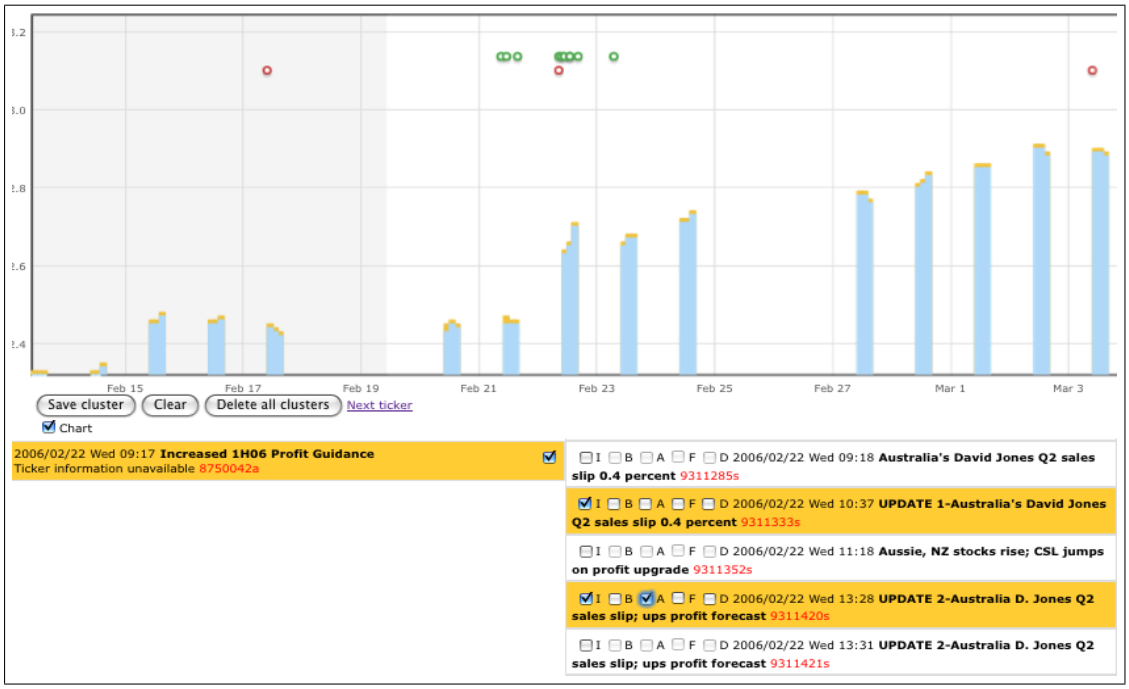

Figure 2: A *screen* from the annotation interface showing links between an announcement and two stories

istic contribution that the RNA story makes. The LINK label applies if the story in an ASX-RNA pair contains information from the announcement. In this case, information is defined as company details that are legally required to be disclosed *first* through an announcement to the ASX (i.e., continuous disclosure). Journalistic contribution mainly concerns the RNA story and can be indicated by any of FIRST, BACK or ANLY. FIRST indicates that the story is the first to cover the information in the announcement. BACK refers to background information regarded as common knowledge and publicly available before the release of the announcement. ANLY describes new information added by the news source, such as analysis, editorial content and new quotes from industry commentators. Table 3 shows examples of the link types. The distinction between BACK and ANLY is subjective since it the annotator must decide whether information is already known (i.e., BACK) or whether it is novel analysis.

The scheme also defines an annotation unrelated to information flow: DIGEST. The Reuters dataset includes stories that contain snippets of news that relate to multiple events and companies, often a daily market report of reviews of 'Hot stocks'. These RNA stories would be likely to report information from many sources and the DIGEST annotation allows their exclusion (although we do not do so in our experiments).

5 Annotation Task

The annotation task and interface were designed to allow annotators to read ASX-RNA pairs and identify any information flow. A pilot annotation team of Finance PhD students assisted in the development of the scheme and interface by participating in a shared annotation task. The initial scheme specified that the links were mutually exclusive but resulted in low Cohen's Kappa agreement scores (Cohen, 1960). After several iterations of relaxing and refining the scheme, Kappa inter-annotator agreement was sufficient to begin the main annotation phase.

A second team of Finance students was hired and after completing a shared task, the seven annotators with consistently high average Kappa score were chosen to continue. An 18 month period from 2005 to mid-2006 was used to create screens. A *screen* consists of the ASX announcements and RNA stories released for a ticker over a fortnight and allows annotators to view the pairs in context and apply information flow links. We randomly sampled three subtasks for each annotator consisting of 215 screens to be completed individually, 50 shared screens and a final 215 individual screens. The shared task midway through the project allowed re-checking of agreement figures and is also used as held-out evaluation data. Due to low agreement, only five annotators' data was used for training and evaluation and not all targeted screens were annotated. Consequently, 1779 individual screens were an-

Task	Mean Kappa
LINK	0.75
FIRST	0.71
BACK	† 0.66
ANLY	† 0.55

Table 4: Mean Kappa inter-annotator agreement scores (N=5). † indicates lower than acceptable Kappa.

Label	Count	%	ASX	RNA
LINK	6,380	4.71	17.1	33.6
FIRST	2,638	1.95	16.7	17.7
BACK	4,707	3.47	15.1	27.2
ANLY	1,759	1.30	7.2	9.9

Table 5: Count and percentages of links in both training and evaluation datasets (135,537 pairs, $lag_{max} = 7\ days$). Percentage of linked documents for each source.

notated for use as training data and 42 shared screens for evaluation.

Figure 2 shows a *screen* from the annotation tool. A screen consists of three main sections: a time-aligned navigation panel spanning the top, then two vertical document lists below. The top timeline panel shows the fortnight of interest and a context week either side (though the right context week is not shown in this figure), showing stories that might be related to the announcements at the edges of the target fortnight.

The panel displays the stock price and rows of dots indicating an ASX announcement or RNA story. In this example, the ASX announcements on the bottom row are followed by a burst of RNA stories on the top row and this visualisation helps annotators to quickly navigate large complex screens with many announcements and stories.

The two time-ordered lists of documents on the bottom of the panel show the ASX announcements on the left and RNA stories on the right. The titles and timestamp are always visible and annotators can click to reveal the RNA story text or open the ASX PDF file. Checkboxes corresponding to the information flow link types are arranged towards the middle of the lists.

Although all evaluation here applies to ASX-RNA pairs, the list presentation format allows annotators to cluster related documents. We define a cluster as an announcement and one or more stories related to that same announcement. The highlighted cluster in Figure 2 consists of the ASX announcement on the left and second and fourth RNA stories on the right hand side. The RNA I checkboxes show cluster membership and the A checkbox shows the ANLY link for the second story. The main benefit of this strategy is efficiency since the top-down view allows annotators to easily isolate clusters without re-reading documents and see which clusters they had previously created.

It is also possible to add multiple ASX announcements to the same cluster, for example when a meeting announcement is followed by a set of presentation slides. However, the main constraint is that clusters be as minimal as possible and any announcements containing new information should form new clusters. For example, a company takeover might span several months of offer and counter-offer but our annotation should pick out the individual stages of the overarching process. Moreover, the minimality constraint encourages conceptual clarity and mirrors the way information is released piecemeal while still allowing later aggregation of clusters if required.

Inter-annotator agreement for our five annotators is assessed using Cohen's Kappa over the shared task of 42 screens. Table 4 shows that acceptable Kappa scores are achieved for LINK and FIRST with borderline Kappa scores for BACK– relative to the threshold at 0.67 (Carletta, 1996). ANLY was annotated with lower agreement and is consistent with annotator feedback during scheme development, where ANLY links were the most difficult to disambiguate from BACK since the distinction between existing and new information proved subjective.

We placed an upper bound of a week on the time lag between ASX and RNA publishing time. Though primarily an optimisation step to reduce the number of pairs for comparison, it is also consistent with the cluster minimality constraint; annotators were encouraged to split clusters that spanned too long a time.

Table 5 shows the count of links (with a maximum lag of a week) across the individual and shared screens and, in the second column, the distribution of the link types. BACK is the most frequent type beyond the prerequisite LINK link, followed by FIRST and then ANLY. While the low ANLY proportion might be due to stories that emphasise topic and background, low Kappa scores for BACK and ANLY makes it difficult to rule out annotator confusion. Table 5 also indicates that while roughly even proportions of documents are linked by FIRST and ANLY, far more RNA stories are linked by LINK and BACK. Indeed, no more than a third of each journalistic contribution link type appear without another (they all co-occur with LINK) , indicating that the annotators applied them with a high degree of overlap.

6 Features

We model the information flow problems using a variety of text similarity and temporal features extracted from the ASX-RNA pairs. Each feature value is binary and real-valued features are placed into equally sized bins (with the exception of lag as mentioned below). The text and title of the announcement and story were

both tokenised using the NLTK's word tokeniser (Bird et al., 2009) and implementation of the Punkt sentence tokeniser (Kiss and Strunk, 2006). Unigram and bigram features are extracted, ignoring punctuation and any n-grams that include a token in NLTK's list of 127 English stopwords.

To model fine-grained textual similarity, we define three set-theoretic classes of bag-of-words features depending on where content is found: intersection (ASX $\cap$ RNA), *only* in the announcement (ASX $\setminus$ RNA) and *only* in the story (RNA $\setminus$ ASX). These methods are applied to unigrams and bigrams in the title and body text of the ASX-RNA pair. The intersection text/title features are used for a baseline approach. The set-theoretic features are mainly designed to model information flow's similarity, and the story's contribution (RNA $\setminus$ ASX).

The information flow problem requires tracking of short units of text such as distinctive terms and figures. We encode this using Text Reuse similarity scores over text unigrams, title unigrams and tokens containing one or more digits (Metzler et al., 2005). The scores calculated are symmetric overlap, asymmetric overlap favouring the RNA story with and without inverse document frequency weighting, a TFIDF overlap score and two cosine similarity scores, one unweighted and one TFIDF weighted.

To capture longer units of reused text, we take tokens *including* stopwords from each sentence and count the number of exact sentence matches between the ASX-RNA pair. Common token sub-sequences are found using Python2.6's difflib library[6]. The sequences include stopwords and have a minimum length of three since we already calculate bigrams. Both the lengths and counts of these sub-sequences are rounded into bins to produce features such as: *seq-len* and *seq-len-count* indicating that there were matches of *len* and that there were *count* of them respectively.

We also extract features to represent the precision of financial figures mentioned in both texts: the more precision used, the more important the figure. For each number string appearing in both texts, if it consists of a non-zero digit followed by any number of zeros or periods, the characters are replaced with `0`. Otherwise, the characters were replaced with `#`. For example, a round number like `5000` would be replaced by `0000` and a more interesting number like `45.3` would be replaced with `####`. The set of these precision-hashed numbers are used as features.

Time is an important factor in news and we propose that the placement of announcements and stories in the news cycle is significant. The temporal features consist of the *time lag* between the release time of the story and the announcement, mapped to bins that increase in size either side of zero (to account for stories that occur before announcements). For example, the bins around zero are: `[-15...-5)`, `[-5...0)`, `[0...5)`, `[5...15)` and are left-closed and right-open so that a pair released at the same time will have a feature value of `[0...5)`. In addition to this, the time of each document release is rounded to the half-hour, generating a feature to represent the ASX's news cycle.

[6] `http://docs.python.org/library/difflib.html`

Label	Training	%	Evaluation	%
Total pairs	30,249	100.0	1,621	100.0
LINK	5,596	18.5	231	14.3
FIRST	2,394	7.9	81	5.0
BACK	4,118	13.6	166	10.2
ANLY	1,472	4.9	72	4.4

Table 6: Distribution of links in the training (30,249 pairs) and evaluation (1,621 pairs) datasets - both use a lag of less than 1 day.

7 Experimental Methodology

The information flow problem is framed as four binary text categorisation tasks over the ASX-RNA pairs – one task for each link type. The development experiments use 10-fold cross validation and we report precision, recall and F-score for classifying pairs as labelled. We do not report scores for classifying *unlabelled* pairs since these are far more common than the labelled pairs. The experiments use the MegaM Maximum Entropy classifier (Daumé III, 2004) with the `binomial` options to represent the binary features.

To compare to human performance, a model is trained using the development data and used to classify the pairs from the shared annotation task. Gold standard *majority* data is created by positing a *true* link where it is marked by a majority of the five annotators – a more difficult task. Each annotator is compared against the majority and the mean result is used as the upper bound on system performance.

Table 5 showed a highly skewed annotation label distribution in the ASX-RNA pairs released within a week of one another. However, approximately 92% of the links occur within 24 hours of one another and Table 6 shows that applying the *short time lag* improves the class distribution. We still consider all pairs in the evaluation dataset, though our system only classifies pairs within the 24 hour lag and thus classifies the 30 LINK labels that lie outside as not linked. Given the lower prior probabilities of true links in the evaluation dataset, we expect the performance to be worse than in development experiments.

Task	Features	P (%)	R (%)	F (%)
LINK	Baseline	85.0	73.1	78.6
LINK	Best	90.9	88.1	89.5
FIRST	Baseline	66.0	43.9	52.7
FIRST	Best	77.0	70.1	73.4
BACK	Baseline	83.4	67.1	74.4
BACK	Best	88.4	83.2	85.7
ANLY	Baseline	78.9	56.0	65.5
ANLY	Best	86.7	75.0	80.4

Table 7: Precision, recall and F-score for cross-fold validation experiments.

8 Results

Table 7 summarises the experimental results, showing baseline and best precision, recall and F-score for *true* link classification. Text intersection unigrams and bigrams, title intersection unigrams and bigrams were used as baseline features and those scores were exceeded for all link types. While higher F-scores were achieved, for the most part, in the tasks with higher prior link probabilities, scores in ANLY were surprisingly high given its low prior of 4.9.

Table 8 shows the best performing (by F-score) feature combinations for each link type. To test the contribution of each feature, subtractive analysis was performed on the best performing feature set for each link type. An experiment is conducted that uses all but one feature and the results compared to best using approximate randomisation (Chinchor, 1995) to assess whether adding the omitted feature results in a statistically significant improvement.[7] Features used are marked with $\cdot$, while features are marked with $\star$ or $\star\star$ if their removal results in significantly worse F-score (at $p<0.05$ and $p<0.01$ respectively).

The first observation to make from the table is that the tasks can be separated into two groups on the set of features that was most successful: LINK/BACK and FIRST/ANLY, though this may also be related to the different prior link probabilities, higher and lower for each group in this case.

Features based on the text play perhaps the broadest role, both modelling information flow and journalistic contribution. Although intersection unigrams and bigrams appeared in all feature sets, text intersection bigrams were only significant in LINK and BACK. One reason might be that they more effectively model topic-level textual similarity while being less susceptible to single words appearing by chance in both texts. The textual similarity measures were significant for the LINK, FIRST and BACK experiments, perhaps because they are able to weight terms more effectively.

[7]We adapt a parsing evaluation script `http://www.cis.upenn.edu/~dbikel/software.html`

Features	LINK	FIRST	BACK	ANLY
ASX$\cap$RNA TEXT-1G	$\cdot$	$\cdot$	$\cdot$	$\cdot$
ASX$\cap$RNA TEXT-2G	$\star\star$	$\cdot$	$\star\star$	$\cdot$
ASX$\setminus$RNA TEXT-1G				
ASX$\setminus$RNA TEXT-2G				
RNA$\setminus$ASX TEXT-1G		$\cdot$		$\cdot$
RNA$\setminus$ASX TEXT-2G		$\cdot$		$\star\star$
ASX$\cap$RNA TITLE-1G	$\cdot$	$\cdot$	$\cdot$	$\cdot$
ASX$\cap$RNA TITLE-2G	$\cdot$	$\cdot$	$\cdot$	$\cdot$
ASX$\setminus$RNA TITLE-1G	$\star\star$		$\star\star$	
ASX$\setminus$RNA TITLE-2G	$\star\star$		$\star\star$	
RNA$\setminus$ASX TITLE-1G	$\cdot$	$\cdot$	$\star\star$	$\cdot$
RNA$\setminus$ASX TITLE-2G	$\star\star$	$\cdot$	$\star$	$\cdot$
TEXT SIMILARITY	$\star\star$	$\star\star$	$\star$	$\cdot$
TITLE SIMILARITY	$\cdot$	$\star\star$	$\cdot$	$\cdot$
SENTENCES	$\cdot$	$\cdot$	$\cdot$	$\cdot$
SEQUENCES	$\cdot$	$\star\star$	$\cdot$	$\cdot$
NUM SIMILARITY		$\cdot$		$\cdot$
NUMBER PRECISION		$\star$		$\cdot$
TIME LAG	$\star\star$	$\star\star$	$\star\star$	$\cdot$
TIME OF DAY	$\cdot$	$\cdot$	$\cdot$	$\star$

Table 8: Feature combinations for the best performing development experiments. Features significant from subtractive analysis are annotated $\star$ ($p<0.05$) and $\star\star$ ($p<0.01$)

Common sequence matching proved significant in detecting FIRST and no other experiments. Reported figures and information are more likely to be reported verbatim, rather than be subject to editing, and this may play a role in the features' success. Of the text set difference features, only bigrams that appeared in the RNA story and not the ASX announcement were significant and only then for ANLY, suggesting that the feature effectively represents commentary. Interestingly, text present only in the ASX announcement was not used in any well-performing experiment. One potential explanation is that the wide variety of text sizes is simply too noisy a feature for the model to generalise.

Titles play an important role in announcements and stories, summarising the event that they report on. Rather than intersection, the ASX and RNA set differences proved to be more significant features. The ASX unigram and bigram varieties of this feature were significant for both LINK and FIRST experiments and the RNA unigrams and bigrams less significant for the same classes. This may indicate that cues of ASX announcement newsworthiness may appear in ASX titles, yet not be repeated in the titles of stories that report on them. Conversely, title terms that indicate that a story reports directly on an announcement may not be found in that announcement's title. In addition to this, titles are often constrained by space and the need for concise communication and are less likely to contain

Task	Baseline	Best	Upper
LINK	62.5	⋆⋆76.7	86.4
FIRST	38.8	53.0	83.1
BACK	56.7	⋆⋆68.0	80.8
ANLY	38.2	45.7	72.5

Table 9: Model F-score agreement with *majority*. Upper is the mean of the F-scores for each annotator and *majority*. ⋆⋆ indicates significance ($p<0.01$)

non-indicative terms. Title similarity was important in LINK and FIRST, the only significant title-based feature for the latter task. Further exploration would be required to measure how much information is transferred in the titles alone.

Numbers are central to information flow in Finance and the two features based on numbers, similarity measures and precision were present in the FIRST and ANLY experiments. The number precision feature was significant for the FIRST experiments, while similarity and precision were just under significance for ANLY. Though these initial results are encouraging, the importance of number to information flow means that more work is required.

Finally, news has a strong temporal dimension and we expected the lag feature to be significant for all link types. While it was for LINK, FIRST and BACK, the time-of-day feature was more significant for ANLY. That the analysis and commentary are the only link types sensitive to their placement in the news cycle points, potentially, to less time critical stories released at regular times.

Table 9 shows performance between the baseline and *majority* in the evaluation task. While the lack of significant results for FIRST and ANLY is somewhat discouraging, reasonable results for LINK and BACK indicate the feasibility of our approach to the information flow problem.

9 Conclusion

Our paper presents a formalisation of the information flow problem in the Finance domain. We present an annotation scheme that codifies flow of facts from primary to secondary sources and apply it to ASX announcements and RNA stories. Moreover, the scheme models three types of journalistic contribution and, despite its difficulty, can be applied with high agreement.

We explore a range of features from diverse fields and combine them to classify the different information flow types. Textual features based on the intersection and differences of document texts and titles prove useful, while number features show promise at identifying financial figures. Temporal features allow modelling of the news cycle and news source responsiveness to identify linked documents.

This paper presents a new approach to the information flow problem in the Finance domain, essentially text categorisation over the *pair* of documents. While bag-of-words performs predictably well in this task, we are able to take advantage of temporal and textual features to classify information flow at 89.5% F-score and journalistic contribution from 73.4% to 85.7% F-score. In evaluation against human performance of 86% F-score, our system scores 77% for flow classification; demonstrating we can feasibly track information flow in Finance text.

10 Acknowledgements

We would like to thank all reviewers, both anonymous and from the Sydney Language Technology Research Group for their feedback. We also thank Dominick Ng, Silvio Tannert, Amy Kwan and Ingo Wiegand at the Capital Markets CRC for their assistance preparing the corpora and managing the annotation project. This research was conducted with the support of a Capital Markets CRC High Achiever's Scholarship.

References

James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998a. Topic detection and tracking pilot study final report. In *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218.

James Allan, Ron Papka, and Victor Lavrenko. 1998b. Online new event detection and tracking. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45, New York, NY, USA. ACM.

ASX. 2008. Continuous disclosure. *ASX Listing Rules*, Chapter 3.

Steven Bird, Edward Loper, and Ewan Klein. 2009. Natural language processing with python.

Sergey Brin, James Davis, and Héctor García-Molina. 1995. Copy detection mechanisms for digital documents. In *SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 398–409, New York, NY, USA. ACM.

Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. *Proceedings of the 29th annual meeting on Association for Computational Linguistics*.

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22:249–254.

Nancy Chinchor. 1995. Statistical significance of muc-6 results. In *MUC6 '95: Proceedings of the 6th conference on Message understanding*, pages 39–43, Morristown, NJ, USA. Association for Computational Linguistics.

Paul Clough, Robert Gaizauskas, Scott S. L. Piao, and Yorick Wilks. 2002. Meter: Measuring text reuse. In *ACL '02: Proceedings of the 40th Annual Meeting on*

Association for Computational Linguistics, pages 152–159, Morristown, NJ, USA. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, April.

Hal Daumé III. 2004. Notes on cg and lm-bfgs optimization of logistic regression. Aug.

William A. Gale and Kenneth W. Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 177–184, Morristown, NJ, USA. Association for Computational Linguistics.

Daniel Gruhl, R. Guha, David Liben-Nowell, and Andrew Tomkins. 2004. Information diffusion through blogspace. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 491–501, New York, NY, USA. ACM.

Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. 1999. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 203–212, College Park, Maryland, June. SIGDAT.

Timothy C. Hoad and Justin Zobel. 2003. Methods for identifying versioned and plagiarized documents. *Journal of the American Society for Information Science and Technology*, 54(3):203–215.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Jon Kleinberg. 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397.

Victor Lavrenko, James Allan, Edward DeGuzman, Daniel LaFlamme, Veera Pollard, and Stephen Thomas. 2002. Relevance models for topic detection and tracking. In *Proceedings of the second international conference on Human Language Technology Research*, pages 115–121, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Chris Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to information retrieval. *Cambridge University Press New York, NY, USA*, Jan.

Donald Metzler, Yaniv Bernstein, W. Bruce Croft, Alistair Moffat, and Justin Zobel. 2005. Similarity measures for tracking information flow. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 517–524, New York, NY, USA. ACM.

Gerard Salton, A Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Ian Soboroff. 2004. Overview of the trec 2004 novelty track. In *The Thirteenth Text Retrieval Conference (TREC 2004)*.

Karen Spärck Jones. 1973. Index term weighting. *Information Storage and Retrieval*, 9(11):619–633.

Michael J Wise. 1996. Yap3: improved detection of similarities in computer program and other texts. *SIGCSE Bull.*, 28(1):130–134.

Akbar Zaheer and Srilata Zaheer. 1997. Catching the wave: alertness, responsiveness, and market influence in global electronic networks. *Manage. Sci.*, 43(11):1493–1509.

Classifying articles in English and German Wikipedia

Nicky Ringland and **Joel Nothman** and **Tara Murphy** and **James R. Curran**
School of Information Technologies
University of Sydney
NSW 2006, Australia
{nicky,joel,tm,james}@it.usyd.edu.au

Abstract

Named Entity (NE) information is critical for Information Extraction (IE) tasks. However, the cost of manually annotating sufficient data for training purposes, especially for multiple languages, is prohibitive, meaning automated methods for developing resources are crucial. We investigate the automatic generation of NE annotated data in German from Wikipedia. By incorporating structural features of Wikipedia, we can develop a German corpus which accurately classifies Wikipedia articles into NE categories to within 1% F-score of the state-of-the-art process in English.

1 Introduction

Machine Learning methods in Natural Language Processing (NLP) often require large annotated training corpora. Wikipedia can be used to automatically generate robust annotated corpora for tasks like Named Entity Recognition (NER), competitive with manual annotation (Nothman et al., 2009). The CoNLL-2002 shared task defined NER as the task of identifying and classifying the names of people (PER), organisations (ORG), places (LOC) and other entities (MISC) within text (Tjong Kim Sang, 2002). There has been extensive research into recognising NEs in newspaper text and domain-specific corpora, however most of this has been in English. The cost of producing sufficient NER annotated data required for training makes manual annotation unfeasible, and the generation of this data is even more important for languages other than English, where gold-standard corpora are harder to obtain.

German NER is especially challenging since various features used successfully in English NER, including proper noun capitalisation, do not apply to German language data, making NEs harder to detect and classify (Nothman et al., 2008). Furthermore, German has partially free word order which affects the reliability of contextual evidence, such as previous and next word features, for NE detection.

Nothman et al. (2008) devised a novel method of automatically generating English NE training data by utilising Wikipedia's internal structure. The approach involves classifying all articles in Wikipedia into classes using a features-based bootstrapping algorithm, and then creating a corpus of sentences containing links to articles identified and classified based on the link's target.

We extend the features used in Nothman et al. (2008) for use with German Wikipedia by creating new heuristics for classification. We endeavour to make these as language-independent as possible, and evaluate on English and German.

Our experiments show that we can accurately classify German Wikipedia articles at an F-score of 88%, and 91% for entity classes only, achieving results very close to the state-of-the-art method for English data by Nothman et al. (2008) who reported 89% on all and 92% on entities only. Nothman et al.'s (2009) NER training corpus created from these entity classifications outperforms the best cross-corpus results with gold standard training data by up to 12% F-score using CoNLL-2003-style evaluation. Thus, we show that it is possible to create free, high-coverage NE annotated German-language corpora from Wikipedia.

2 Background

The area of NER has developed considerably from the Message Understanding Conferences (MUC) of the 1990s where the task first emerged. MET, the Multilingual Entity Task associated with MUC introduced NER in languages other than English (Merchant et al., 1996) which had previously made up the majority of research in the area. The CoNLL evaluations of 2002 and 2003 shifted the focus to Machine Language, and further multilingual NER research incorporated language-independent NER. CoNLL-2002 evaluating on Spanish and Dutch (Tjong Kim Sang, 2002) and CoNLL-2003 on English and German (Tjong Kim Sang and De Meulder, 2003).

The results of the CoNLL-2002 shared task showed that whilst choosing an appropriate machine learning technique affected performance, feature choice was also vital. All of the top 8 systems at CoNLL-2003 used lexical features, POS tags, affix information, previously predicted NE tags and orthographic features.

The best-performing CoNLL-2003 system achieved an F-score of 88.8% on English and 72.4% on German (Florian et al., 2003). It combined Maximum Entropy Models and robust risk minimisation with the use of external knowledge in the form of a small gazetteer of names. This was collected by manually browsing web pages for about two hours and was composed of 4500 first and last names, 4800 locations in Germany and 190 countries. Gazetteers are very costly to create and maintain, and so considerable research has gone into their automatic generation from online sources including Wikipedia (Toral and Muñoz, 2006).

The CoNLL-2003 results for German were considerably lower than for English, up to 25% difference in F-score (Tjong Kim Sang and De Meulder, 2003). The top performing systems all achieved F-scores on English more than 15 higher than on German.

2.1 German NER

German is a very challenging language for NER, because various features used in English do not apply. There is no distinction in the capitalisation of common and proper nouns so the number of word forms which must be considered as potential NEs is much larger than for languages such as English. German's partially free word order also means that surface cues, such as PER entities often preceding verbs of communication, are much weaker.

A final consideration is gender. The name Mark is likely to be on any list of German person names, but also makes up part of Germany's old currency, the Deutsche Mark, also known as D-Mark or just Mark (gender: female; *die*), and also has the meaning 'marrow' (gender: neuter; *das*). Whilst gender can sometimes disambiguate word senses, in more complicated sentence construction, gender distinctions reflected on articles and adjectives can change or be lost when a noun is used in different cases.

2.2 Cross-language Wikipedia

The cross-lingual link structure of Wikipedia represents a valuable resource which can be exploited for inter-language NLP applications. Sorg and Cimiano (2008) developed a method to automatically induce new inter-language links by classifying pairs of articles of two different langauges as connected by a inter-language link. They use a classifier utilising various text and graph-based features including edit distance between the title of articles and link patterns. They find that since the fraction of bidirectional links (cases where the English article e.g. Dog is linked to the German article Hund which is linked to the original English article) is around 95% for German and English, they can be used in a bootstrapping manner to find new inter-language links. The consistency and accuracy of the links was also found to vary, with roughly 50% of German language articles being linked to their English equivalents, and only 14% from English to German.

Richman and Schone (2008) proposed a system in which English Wikipedia article classifications are used to produce NE-annotated corpora in other languages, achieving an F-score of up to 84.7% on French language data, evaluated against human-annotated corpora with the MUC evaluation metric. So far there has been very little research into directly classifying articles in non-English Wikipedias.

2.3 Learning NER

Machine learning approaches to NER are flexible due to their statistical data-driven approach, but training data is key to their performance (Nothman et al., 2009). The size and topical coverage of Wikipedia makes its text appropriate for training general NLP systems.

The method of Nothman et al. (2008) for transforming Wikipedia into an NE annotated corpus relies on the fact that links between Wikipedia articles often correspond to NEs. By using structural features to classify an article, links to it can be labelled with an NE class.

The process of deriving a corpus of NE annotated sentences from Wikipedia consists of two main sub-tasks: (1) selecting sentences to include in the corpus; and (2) classifying articles linked in those sentences into NE classes. By relying on redundancy, articles that are difficult to classify with confidence may simply be discarded.

This method of processing Wikipedia enables the creation of free, much larger NE-annotated corpora than have previously been available, with wider domain applicability and up-to-date, copyright free text. We focus on the first phase of this process: accurate classification of articles.

NLP tasks in languages other than English are disadvantaged by the lack of available data for training and testing. Developing more automated methods of language-resource generation which is independent of existing data sets is an important and challenging goal. We work towards generating high-coverage training corpora which can be used for a range of German NLP.

3 Data

To learn a classification of German Wikipedia articles, we labelled a corpus of English Wikipedia articles. Wikipedia's inter-language links allow us to then develop classifiers for all articles in English and German (or other language) Wikipedias. We use XML dumps of Wikipedia from March 2009 for both languages.

3.1 Article selection

Both Nothman et al. (2008) and Dakka and Cucerzan (2008) have labelled collections of Wikipedia articles with gold standard classifications. Both of these consist of randomly selected articles, Dakka and Cucerzan's consisting of a random set of 800 pages, expanded by list co-occurrence. Nothman et al.'s data set initially consisted of 1100 randomly sampled articles from among all Wikipedia articles. This biased the sample towards entity types that are frequent in Wikipedia, such as authors and albums, but poorly represented countries, for example, which are important but are only a small proportion of Wikipedia's articles. A high number of the selected articles were stubs or other pages which were comparatively underdeveloped in structure and text. As a result, the data set was augmented with a further 200 articles, randomly sampled from among articles with at least 700 incoming links (*in-links*).

Rank	Article	Pageviews
1	2008 Summer Olympics	4 437 251
2	Wiki	4 030 068
3	Sarah Palin	4 004 853
4	Michael Phelps	3 476 803
5	YouTube	2 685 316
6	Bernie Mac	2 013 775
7	Olympic Games	2 003 678
8	Joe Biden	1 966 877
9	Georgia (country)	1 757 967
10	The Dark Knight (film)	1 427 277

Table 1: Most frequently viewed Wikipedia articles from August 2008, retrieved from http://stats.grok.se

Rank	Title	Inlinks
1	United States	543 995
2	Australia	344 969
3	Wikipedia	272 073
4	Association Football	241 514
5	France	227 464

Table 2: Most linked-to articles of English Wikipedia.

We took a more complex approach to choosing articles for inclusion in our data set, to ensure greater utility for multilingual Wikipedia tasks. We selected ∼2300 articles from:

- the top 1000 most frequently viewed, based on August 2008 statistics (see Table 1), and
- the most linked-to articles (see Table 2),

with the constraint that they appear in at least the top 10 largest language Wikipedias (Table 3).

Wikipedia	Articles
English	3 500 000
German	950 000
French	850 000
Polish	650 000
Japanese	650 000
Italian	600 000
Dutch	550 000
Spanish	500 000
Portuguese	500 000
Russian	450 000

Table 3: Top ten Wikipedia languages by number of articles (nearest 50 000) as at September 2009.

Dataset	# articles	Paras	Sents	Cats
English 0805	1 296	3.0	36.4	4.6
English 0903	2 269	8.8	122.4	6.4
German 0903	2 269	4.8	84.1	3.3

Table 4: Average size (in paragraphs, sentences and categories) of Nothman et al.'s ∼1300 labelled articles from 2008 and our ∼2300 articles from March 2009.

We experimented with selecting the articles with the most inter-language links, but results were not meaningful; languages such as Volapuk may have fewer than 30 speakers, but more than 100,000 articles, most of which are stubs created and edited automatically. Reducing the languages of interest to 10 allowed us to focus on selecting more meaningful articles, using the criteria above. Although we deemed these criteria appropriate, they skew the corpus to events relevant in August 2008; the Summer 2008 Olympics, upcoming American Presidential Election and conflict between Russia and Georgia were prominent in the data.

In Table 4, we show that we succeed in selecting articles which are more substantial than the random sample of Nothman et al. (2008). Our method largely avoided the "long tail" of more obscure articles, such as old songs, sports players or archaeological finds, whose representation in a random sample is disproportionate to their utility.

3.2 Annotation

Our corpus was created by manually classifying approximately 2300 English articles which had German equivalents, selected as described in section 3.1 using a custom annotation tool described in Tardiff et al. (2009). It allowed for an arbitrary number of annotators, and for multiple annotations to be compared.

PER	LOC	ORG	MISC	NON	DIS	Total
271	648	229	392	650	79	2 269
12%	29%	10%	17%	29%	3%	100%

Table 5: Breakdown of manual classifications: People, Locations, Organisations, Miscellaneous, Common and Disambiguation.

Annotation was carried out using a hierarchical fine-grained tag-set based upon guidelines from the BBN Technologies' 150 answer types (Brunstein, 2002). Categories were able to be added into the hierarchy and either 'grown' or 'shrunk' to better fit the data as the annotators saw it. The ability to add categories is especially important when annotating Wikipedia because many categories such as types of works of art or products are not adequately covered in BBN.

The corpus was annotated using fine-grained categories, adding more information for use in future work, and enabling easier annotation, as they allow an annotator to classify a topic into a well-defined sub-category, which can then be uniformly mapped to a coarse-grained category. For example, all hotels can be classified as HOTEL, which then can be mapped to either ORG or LOC as decided after annotation.

The annotation process allowed for a high level of feedback to annotators, with statistics including inter-annotator agreement and a list of articles not uniformly classified available during annotation. This allowed annotators to quickly and easily identify digressions from one another.

All articles were double-annotated. After tagging the first 78 articles, we discussed conflicts and refined the annotation scheme. The two annotators then both classified a further 1100 articles each, achieving inter-annotator agreement of 97.5% on fine-grained tags, and 99.5% on coarse-grained tags. A further discussion and annotation round of the remaining ∼1100 followed, and the final inter-annotator agreement was 99.7% on fine-grained tags and 99.9% on coarse-grained tags, creating a highly accurate corpus which we plan to release upon publication. Coarse-grained class distribution is given in Table 5.

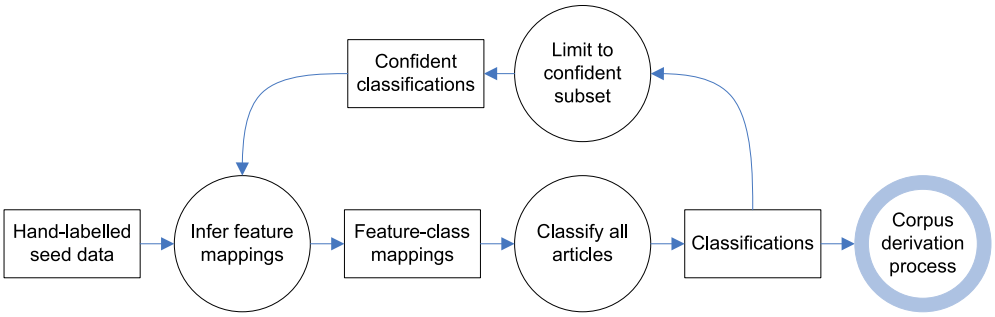

Figure 1: A bootstrapping approach to article classification

4 Classification

Classification of Wikipedia's articles into semantic groupings is useful for applications such as named entity recognition (Kazama and Torisawa, 2007; Nothman et al., 2008) and ontology construction (Suchanek et al., 2007). The Wikipedia category hierarchy is a folksonomy and not directly suitable for NLP tasks. Instead, rule-based (Toral and Muñoz, 2006; Richman and Schone, 2008), semi-supervised (Nothman et al., 2008; Mika et al., 2008) and supervised (Dakka and Cucerzan, 2008) article classifiers have derived coarse-grained entity groupings or taxonomies.

Features used in classification are varied. Suchanek et al. (2007) used Wikipedia categories to map articles to WordNet, but noted that conceptual categories in English usually have plural head nouns (e.g. COASTAL **CITIES** IN AUSTRALIA) which describe the nature of member articles, as opposed to thematic categories like JAMES BOND. Richman and Schone (2008) scanned the hierarchy of categories for known phrases to classify articles into named entity categories.

Since an article's topic is usually defined in its first sentence, Toral and Muñoz (2006) try to match words from the opening sentence to a related class through the WordNet taxonomy. The specific use of the predicative head noun following a copula (is, were, etc.) in the first sentence was suggested by Kazama and Torisawa (2007) as a single feature by which articles may be grouped.

Other approaches utilise the co-occurrence of entities in lists (Watanabe et al., 2007; Bhole et al., 2007; Dakka and Cucerzan, 2008); presence of entities in particular fields of *infobox* templates which summarise the properties and relations of article topics (Mika et al., 2008); and bag-of-words SVM classification (Dakka and Cucerzan, 2008; Bhole et al., 2007).

Although using different data sets, both Nothman et al. (2008) and Dakka and Cucerzan (2008) have reported F-scores of approximately 90% for classification into CONLL style entity categories.

4.1 Classifying Wikipedia articles

Nothman et al. (2008)'s bootstrapping classifier works as follows (see Figure 1): By initially associating features of each training instance with its gold-standard class label, an initial classification of all articles in Wikipedia is produced. Features that are consistently associated with a particular predicted class are then mapped to that class, including those not present in the hand-labelled data. These classification and mapping stages are then repeated, increasing feature coverage until the classifications are generally stable. Such an approach allows for high recall over sparse multi-valued features like the Wikipedia category membership of each article. We extend their approach to German Wikipedia.

4.2 Increasing non-entity recall

The following rules help to determine whether an article describes a named entity, or a non-entity topic (NON).

Capitalisation In English, all named entities are proper nouns, which are conventionally capitalised. This can be utilised by observing the capitalisation of all incoming links, with basic features allowing for determiners and non-conventional orthographies such as gzip or iPod.

In German, since all nouns are capitalised, this distinction is lost. Furthermore, adjectival forms

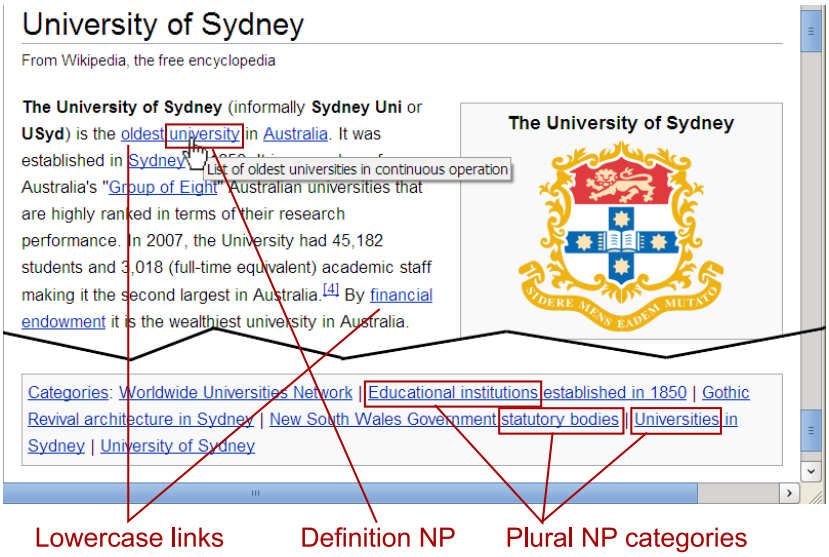

Figure 2: A portion of the Wikipedia article on the University of Sydney with some useful features marked.

including NEs of countries (eg. "Australian") are not capitalised in German ("australisch"), which means even a basic heuristic to check whether a link is a noun is not feasible.

List identification If the English article title begins List of or German, Liste, we mark it as NON.

Disambiguation identification Wikipedia's disambiguation articles list candidate referents for a particular title. The German page Mark lists amongst others, the name, substance (marrow), river, saints and various British tanks from WW1 of the same name. Most disambiguation pages are children of a category of DISAMBIGUATION, many have the word Disambiguation or Begriffsklärung in the title, and further information is available in the form of disambiguation templates.

4.3 Bootstrapped features

For general classification, we extracted features from articles, which may each be mapped to an entity class. These mappings are produced by the bootstrapping process.

Category nouns Head nouns from Wikipedia category titles for both English and German, extracted using C&C tools (Curran and Clark, 2003) in English and the Tree-Tagger in German (Schmid, 1995) to POS-tag and chunk the sentences. In English, the category feature only applied to plural head nouns (and bigrams thereof) following Suchanek et al.'s (2007) suggestion that these best represent ontology. Differences in both language and the structure of the German Wikipedia project invalidate this approach in German: conceptual categories are not plural, and forms that are bigrams in English are generally compound nouns. Hence we experimented with ASV toolbox (Chris Biemann and Holz, 2008) to extract a head morpheme. This allows PREMIERMINISTER (Prime Minister) and WISSENSCHAFTSMINISTER (Science Minister) to both be interpreted as MINISTER, and KERNBRENNSTOFFAUFBEREITUNGSANLAGE (nuclear fuel treatment facility) to become ANLAGE (facility).

Definition nouns We term a *definition noun* to be the first noun following a copula in the first sentences, such as university in Figure 2. Definition nouns are extracted using POS-tagging and chunking as per category nouns, from articles which had been split into sentences and tokenised according to the method described in Nothman et al. (2008).

For each article, the number of category nouns mapped to each class is counted, and the most frequent class is used to label the article. If this is inconclusive, or the highest class leads by only one category, the definition noun is used to decide the class. Where there are no mapped category nouns or definition nouns available, or no winning class can be decided, the article class is marked as unknown (UNK).

An article classification is considered confident for use in bootstrapping if it is not labelled UNK, and if none of the features disagree (i.e. all category and definition features available map to the same class).

Iter	German 0903			English 0903			English 0805		
	P	R	F	P	R	F	P	R	F
0	93	79	85	95	88	92	93	73	82
1	93	83	88	97	91	94	93	79	85
2	93	84	88	95	90	93	93	80	86
3	93	84	88	95	90	93	95	84	89

Table 7: Results of bootstrapping iterations on the held-out test set of German and English, compared to English 0805, as reported in Nothman (2008).

5 Results

We present results for our German Wikipedia classifier, exploring the effect of bootstrapping and feature variants in comparison to Nothman et al.'s (2008) English Wikipedia classifier.

We were able to achieve 88% F-score for German classification on a held-out test-set of 15% of the data (Table 6). These results are comparable to those presented by Nothman on English, but slightly lower than those using our larger annotated training corpus on a 2009 dump of English Wikipedia.

Data Validation The inter-language links between the German and English Wikipedias were checked and found to be reliable, with only two errors in links from English to German pages from the test set used for experimentation, which is consistent with the findings of Sorg and Cimiano (2008). Both of these were articles pointing to disambiguation pages: Nikon (describing the company) and Isosceles (describing the type of triangle). In the held-out training set, similar, though few, mis-links were found: the English article on French playwright Molière linking to Molière (1978), a French film depicting his life, and the German article Ryan Vikedal, a former member of the band Nickelback, links to the English article of the same name, which itself redirects to Nickelback. It should be noted that all of these examples were correctly classified by our process. When these errors were corrected, F-score improved by under 0.1%, showing that even with occasional noise, inter-wiki language links can be used to produced good-quality data. The results we present use a uncorrected test set.

Bootstrapping Bootstrapping was found to be less effective than in Nothman (2008) (see Table 7), where it was more needed to increase recall given less manually-labelled seed data. With the larger seed, bootstrapping proved more important on the German data than English, with recall increasing 5% compared to 2%, still falling short of the 11% increase found by Nothman. In our experiments, we found that the results were unchanging after the second feedback stage.

Feature Analysis In Table 6, we examine the effects of removing some classification features, and compare against the same process on English Wikipedia. In English, the capitalisation feature improves recall slightly, as opposed to the substantial increase found in Nothman's work; we might expect German, in which capitalisation is not used, to be disadvantaged by a similar amount.

Category nouns are seen to be by far the most important feature, especially in German. Our experiments to extract the morphology-based head from each category noun were an attempt to increase recall. We observed a slightly higher recall in the seed classification, but the bootstrapping process – also designed to improve recall – was more effective with the finer granularity of whole category noun features. This ultimately led to slightly reduced recall, leading us to use whole category nouns in our remaining experiments.

Definition nouns gave mixed results. In German they improved recall but had little effect on precision, while in English they improved precision and recall.

Cross-validation The results of ten-fold cross-validation are shown in Table 8, with a class breakdown. Our system left 8% of German and 6% of English Wikipedia articles unclassified (UNK). Nothman (2008) reports that 10% of articles were left unclassified. Our present work was able to classify a greater proportion due to our selection of more, higher-quality seed articles.

The German system performs very well on LOC and on MISC, which is known to be difficult to classify, achieving almost equivalent scores to English. The system also achieves a high F-score on PER. All of the false negatives when classifying people were on articles describing fictional characters such as Luzifer, Godzilla and Hermaphroditos. The error analysis of ORG also shows that we fail to correctly classify articles which the annotators also were unsure of, such as eBay and amazon.com, and Jedi. MISC often appeared incorrectly classified as ORG, showing the often blurred distinction between a product and the organisation which produces it (eg: Jeep and Airbus A380). The BBN guidelines also proved difficult for the classifier to adhere to, with 'attractions' such as the Nürburgring being classified as LOC not MISC.

Table 9 compares the precision, recall and F-score of English and German overall and on entity classes only. We also report the standard deviation of performance over the ten folds of cross-validation. The larger gap between all-class and entity class results in German reflects the low NON recall (76% as opposed to 90% in English), likely due to no available capitalisation feature.

Classification features	German 0903			English 0903			English 0805		
	P	R	F	P	R	F	P	R	F
All features	-	-	-	95	90	93	95	84	89
− Capitalisation	93	84	88	96	89	92	92	80	85
+ Category morphology	93	83	88	-	-	-	-	-	-
− Definition nouns	93	81	87	93	88	91	95	80	87
− Category nouns	48	7	12	76	28	41	48	13	21

Table 6: Subtractive feature analysis on the held-out test set, comparing German Wikipedia with English (0903) performance, and the results reported by Nothman (2008) (English 0805).

Wikipedia		P	R	F
English 0903	All	94 ±2	89 ±1	91 ±1
	Entities	98 ±1	89 ±2	93 ±1
German 0903	All	91 ±3	84 ±3	88 ±2
	Entities	97 ±2	87 ±4	92 ±3

Table 9: Classification performance (average and standard deviation) over ten-fold cross-validation.

Class	%	German 0903			English 0903		
		P	R	F	P	R	F
NON	29	85	76	80	86	90	88
DAB	3	92	99	96	100	90	95
LOC	29	98	95	96	99	97	98
MISC	17	89	71	78	97	67	79
ORG	10	93	85	89	97	91	94
PER	12	92	94	93	96	98	97

Table 8: Class distribution of manually classified articles and average results of ten-fold cross-validation.

6 Conclusion

Our work develops a semi-supervised classifier assigning German Wikipedia articles to named entity tags, in comparison to English Wikipedia. In doing so, we labelled a large corpus (2269 articles) of English Wikipedia pages, and validated the use of Wikipedia's inter-language links to transfer those training classifications to the smaller German encyclopedia.

In distinction from previous annotations of Wikipedia data, we produced a corpus with fine-grained classes, extending on BBN's 150 answer types (Brunstein, 2002), and consisting of only articles which satisfy popularity criteria.

The classifier we have produced for German Wikipedia achieves very high precision (97%) and recall (87%) on entity classes. Due to differences between English and German language, orthography and Wikipedia editorial style, we had to modify the semantic and structural features previously used to classify English Wikipedia articles (Nothman et al., 2008). Our use of bootstrapping to spread this semantic knowledge to features unseen in training greatly improves performance in German, in which capitalisation features cannot be easily applied to distinguish NEs from non-entities, and in which there are fewer features available for classification, due to a smaller, less-developed Wikipedia.

We intend to improve the classifier by exploring further features, as well as the integrity of article resolution and inter-language links.

The results we have presented in German are only 3% F-score lower than on English articles and 1% F-score lower when only evaluating on NEs. The CoNLL-2003 shared task presented a 12% minimum reduction in performance for German NER when compared to English (Tjong Kim Sang and De Meulder, 2003). This substantial difference is due either to the difficulty of the NER task in German, or to the paucity of training data available in the CoNLL-2003 shared task, where the German training data marked only half as many NEs as the English corpus. By transforming the links in Wikipedia into entity annotations, we intend to generate large NE-annotated corpora, and to evaluate their use for learning German NER. Our high-accuracy classifier therefore reduces the need for expensive manual annotation in languages other than English where resources tend to be scarce.

Acknowledgments

We would like to thank members of the Language Technology Research Group and the anonymous reviewers for their helpful feedback. This work was partially supported by the Capital Markets Cooperative Research Centre Limited (CMCRC) and by Australian Research Council Discovery Project DP0665973. Joel Nothman was supported by a University of Sydney Vice-Chancellor's Research Scholarship and a CMCRC PhD Scholarship.

References

Abhijit Bhole, Blaž Fortuna, Marko Grobelnik, and Dunja Mladenić. 2007. Extracting named entities and relating them over time based on Wikipedia. *Informatica*, 31:463–468.

Ada Brunstein. 2002. Annotation guidelines for answer types. LDC2005T33, Linguistic Data Consortium, Philadelphia.

Gerhard Heyer Chris Biemann, Uwe Quasthoff and Florian Holz. 2008. Asv toolbox: a modular collection of language exploration tools. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 1760–1767, Marrakech, Morocco, May. European Language Resources Association (ELRA).

James R. Curran and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of the seventh conference on Natural Language Learning*, pages 164–167, Morristown, NJ, USA.

Wisam Dakka and Silviu Cucerzan. 2008. Augmenting Wikipedia with named entity tags. *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pages 545–552.

Radu Florian, Abe Ittycheriah, Hongyang Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 168–171.

Jun'ichi Kazama and Kentaro Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707.

Roberta Merchant, Mary Ellen Okurowski, and Nancy Chinchor. 1996. The multilingual entity task (MET) overview. In *Proceedings of the Tipster Text Program Phase II*, pages 445–447, Vienna, Virginia, May.

Peter Mika, Massimiliano Ciaramita, Hugo Zaragoza, and Jordi Atserias. 2008. Learning to tag and tagging to learn: A case study on Wikipedia. *IEEE Intelligent Systems*, 23(5):26–33.

Joel Nothman, James R. Curran, and Tara Murphy. 2008. Transforming wikipedia into named entity training data. In *Proceedings of the Australian Language Technology Workshop*, pages 124–132.

Joel Nothman, Tara Murphy, and James R. Curran. 2009. Analysing Wikipedia and gold-standard corpora for NER training. *In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 612–620, March.

Joel Nothman. 2008. *Learning Named Entity Recognition from Wikipedia*. School of Information Technologies, University of Sydney, Honours Thesis.

Alexander E. Richman and Patrick Schone. 2008. Mining wiki resources for multilingual named entity recognition. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1–9.

Helmut Schmid. 1995. Improvements in Part-of-Speech Tagging with an Application to German. *Proceedings of the ACL-SIGDAT Workshop*, March.

Philipp Sorg and Philipp Cimiano. 2008. Enriching the crosslingual link structure of wikipedia classification-based approach. *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artifical Intelligence*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge — unifying WordNet and Wikipedia. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, Banff, Alberta, Canada.

Sam Tardiff, James R. Curran, and Tara Murphy. 2009. Improved text categorisation for Wikipedia named entities. In *Proceedings of the Australian Language Technology Workshop*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 142–147.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 1–4.

Antonio Toral and Rafael Muñoz. 2006. A proposal to automatically build and maintain gazetteers for named entity recognition using Wikipedia. In *Proceedings of the Workshop on NEW TEXT, 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. 2007. A graph-based approach to named entity categorization in Wikipedia using conditional random fields. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 649–657.

The AusNC Project: Plans, Progress and Implications for Language Technology

Simon Musgrave
Linguistics Program
Monash University
VIC 3800 Australia
Simon.Musgrave@arts.monash.edu.au

Michael Haugh
School of Languages and Linguistics
Griffith University
QLD 4111 Australia
m.haugh@griffith.edu.au

Abstract

In the last eighteen months, a consensus has emerged from researchers in various disciplines that a vital piece of research infrastructure is lacking in Australia, namely, a substantial collection of computerised language data. A result of this consensus is an initiative aimed at the establishment of an Australian National Corpus. The progress of this initiative is presented in this paper, along with discussion of some important design issues and a consideration of how the initiative relates to the field of language technology in Australia.

1 Introduction

Large-scale corpora are becoming an increasingly important resource in language research, including many sub-disciplines within language technology. An initiative has developed over the last year or more which aims to construct such a corpus as a key element of research infrastructure for Australia. A national repository of language data would have significant value as research infrastructure for a number of research communities in Australia and overseas, thereby increasing access to Australian language data and widening the global integration of research on language in Australia. It would facilitate collaborative ventures in collecting new language data to support multimodal research in human communication and it would consolidate presently scattered and relatively inaccessible collections of historical language data where possible within the Australian National Corpus (AusNC). Such data is of interest not only to researchers in linguistics and applied linguistics, but also to members of the wider Humanities and Social Sciences and informatics research communities who have an interest in Australian society. Such a large annotated language dataset would also provide invaluable training data for work in natural language processing, speech recognition, and the further development of semi-automated annotation.

In this paper, we will give a short overview of the progress of the initiative to date. This will be followed by an introduction to some design questions which have been the subject of discussion in the preliminary phase of the project, and a consideration of some implications of the project for the field of language technology.

2 History and Recent Progress

At least two projects can be considered to have made substantial contributions to corpus-building in Australia before the present initiative was launched. From 1986, the Australian Corpus of English was compiled at Macquarie University.[1] This corpus consists of 500 text samples of (minimally) 2,000 words each, giving a total size of approximately 1,000,000 words. This corpus has been integrated into the International Corpus of English project. The Australian National Database of Spoken Language was collected in the years between 1991 and 1995.[2] This corpus consists of recordings of various types of spoken language plus associated transcripts. Although some of the material in this collection is taken from pairs of speakers collaborating on a map task, the majority of the corpus consists of recordings of speakers reading carefully chosen prompts. The number of native speakers of Australian English who were recorded is 108. Additionally, 96 speakers from two migrant groups were recorded for a subset of the material, and a speaker from each of nine other migrant groups was also recorded. Given the nature of the material, it is not useful to attempt an estimate of the size of this corpus in terms of number of words.

Since more than a decade has now passed since these two corpora were constructed, members of various language-based research communities in Australia have come to see the importance of establishing a national corpus as an essential aspect of research infrastructure. Although primary interest in

[1] Australian Corpus of English: < http://khnt.hit.uib.no/icame/manuals/ace/INDEX.HTM> International Corpus of English: < http://ice-corpora.net/ice/index.htm>
[2] ANDOSL: < http://andosl.anu.edu.au/andosl/>

such a resource would come from linguists and applied linguists, it was also clear that a number of other groups of researchers would derive value from such a resource. In July 2008, the Australian Linguistic Society (ALS) and the Applied Linguistics Association of Australia both held their annual conferences in Sydney, and the opportunity was taken to hold a meeting of scholars interested in the development of a national corpus. The outcome of this meeting was a "Statement of Common Purpose" which includes the following wording:[3]

> the aim of developing a freely available national corpus is that it can become an ongoing resource not only for linguists, but also historians, sociologists, social psychologists, and those working in cultural studies with an interest in Australian society or culture. We therefore see such a corpus as an important part of the development of research infrastructure for humanities researchers in Australia.

The initial list of signatories to this statement has expanded since that meeting and now has 45 names on it.

The existence of the Statement of Common Purpose and of the demonstrated support for it allowed the leaders of the initiative to approach the Australian Academy of the Humanities and the ARC Network in Human Communication Science (HCSNet) to seek funding. This approach was successful, and has made possible an initial phase of planning activity. Firstly, a workshop entitled 'Designing the Australian National Corpus' was held in December 2008 as part of the HCSNet Summerfest 2008. Selected papers from this workshop will appear shortly (Haugh et al, in press [2009]). A second workshop supported by HCSNet will be held (at the same time as the ALTA 2009 Workshop) and this meeting will concentrate on questions about data sources and tools.

Another workshop, supported by the Australian Academy of the Humanities ,was held in Brisbane in May 2009 concentrating on legal and ethical issues. As discussed in the following section, the current plan for AusNC is that the corpus will include significant amounts of material from the World Wide Web and other types of computer-mediated communication. But issues of copyright and, in some cases, of confidentiality arise in relation to such data (Lampert, in press [2009]), and these issues must be resolved before data collection can begin. There may also be confidentiality and copyright problems in making available existing data which has not previously been openly accessible.

The Statement of Common Purpose discussed previously was the outcome of a meeting of interested parties and not a formal activity of the learned societies from which those parties were drawn. However, the 2009 meeting of the ALS committed the society's formal support to the initiative, with the following statement appearing in the minutes of the annual general meeting:

> The meeting expressed its strong support for this initiative to develop an Australian National Corpus, which will stand out as a significant national resource and which will contribute to the research strength of this country.[4]

In addition, that meeting voted to contribute $2500 to the initiative to support the conduct of an audit of existing language data in Australia.

This audit will commence in late 2009 and will have several aspects to it. Firstly, a survey will be sent to individuals and organizations which might be expected to have relevant holdings of data, such as linguistics departments of universities and other research bodies. Secondly, contact will be made with bodies which are known to have significant holdings such as the Australian Broadcasting Corporation and the National Film and Sound Archive. And finally, information about privately held data will be sought by making a request in the mass media.

These various activities are being directed by a steering committee which was formed following the workshop in December 2008. A list of the members of this committee can be found in the Appendix to the current paper.

3 Planning the AusNC

Initial discussions concerning a possible AusNC have emphasized the diversity of research agendas which it might support and the corresponding diversity of content which might be desirable. In this section, we will present some of the issues which have been raised in these discussions, concentrating on three areas. Firstly, there is a consensus that an AusNC must have a carefully planned core component which is comparable to other large corpora, but questions remain about whether technological change should influence this design. Secondly, there is also consensus that an AusNC should represent language use in Australia beyond Australian English, which would make it significantly different from existing national corpora. Thirdly, if an AusNC is to accomplish the various goals mentioned here, it is clear that the design of the technical infrastructure will be of great importance.

[3] The full text is available at http://blogs.usyd.edu.au/elac/2008/08/australian_national_corpus_ini.html

[4] < http://www.als.asn.au/newsletters/alsnews200908.html>

3.1 Core corpus design

A corpus is planned 'to represent a language or some portion of a language' (Biber, Conrad and Reppen, 1998: 246). In the case of an AusNC, one intention is to represent the English language as used in Australia. However, it would not be sensible to attempt to achieve this goal without taking into account comparable existing corpora. One possible strategy is that adopted by the International Corpus of English project, which has one basic design which is followed as closely as possible by all the contributing sub-corpora (Nelson, 1996). An Australian component of ICE already exists, as discussed in section 2, but the ambition of the AusNC project is to achieve a corpus which is bigger than that (1 million words) by at least an order of magnitude. The benchmark for comparability then becomes either the British National Corpus (BNC, Leech, 1992) or the American National Corpus (ANC, Ide and Macleod, 2001). These two corpora are not identical in design; although ANC was initially based on the design of the BNC, it has diverged in the course of its development. Therefore if direct comparability is sought, it is necessary to make a choice between these two. BNC is recognised as a crucial project in the history of corpus linguistics, but it is also now almost twenty years old and therefore has limitations which will be discussed shortly. ANC is also not an ideal model, as its design has evolved over time in response to various pressures (Ide, in press [2009]).

The design of the AusNC has not yet been finalized, but there is little doubt that it will include a very substantial body of text data which can be utilised for comparison with sub-corpora of the BNC or the ANC. Nevertheless, questions remain about the extent to which it is sensible to make comparability a high priority. In particular, the BNC was assembled around 1990, and therefore computer-based text types are scarcely represented in it. Any attempt to represent the use of the English language in Australia in the first decades of the 21st century obviously cannot afford to neglect such genres, and the AusNC initiative can be expected to include substantial amounts of such data. But should this be seen as an aspect of the corpus additional to those sections which provide comparability with earlier collections, or should some elements of comparability be sacrificed in order to make coverage of the newer genres more complete? Inevitably, such decisions will in the end be questions about resource allocation, but the decisions will have to be made relative to the expressed needs of various research communities.

The development of computer-mediated communication and the recognition of computer-based textual genres is one important change since the time when the BNC was assembled. Another is the huge improvement in the possibilities for creating and disseminating high-quality recordings, both audio and video, of language in use (see, for example, Thieberger and Musgrave, 2007). Concurrent with these developments, and interdependent with them, has been an increasing focus on multimodal data as the basis for comprehensive language research and this change is in turn interdependent with the emergence of language documentation as a sub-field of linguistics (Haugh, in press [2009], Musgrave and Cutfield, in press [2009]). A major corpus being designed now must take these developments into account, and this means that the AusNC will very likely include a substantial component of recordings of actual language use of various types. For such material, the actual multimodal material will be the basic data, in contrast to the approach of the BNC, which includes approximately 10% of data from spoken language, but only transcripts are immediately accessible for analysis; the original sound recordings are part of the Sound Archive of the British Library, but are not treated as a part of the corpus itself. The proposed inclusion of audio(visual) recordings and computer-mediated communication in AusNC inevitably means that at least part of the language data held in the corpus will not directly comparable with other major corpora (see section 3.2), but this, on the other hand, raises extremely interesting research possibilities (see section 4).

3.2 Other material

AusNC has as one of its aims to represent language in Australia in total, that is, to go beyond only representing the use of (more or less) standard English in Australia. This aim is of considerable importance to many members of the research communities involved in the initiative, and can be considered a core objective. Australia was a site of great linguistic diversity before European settlement (Dixon 2002). A small part of that diversity remains and the indigenous people of Australia also speak distinctive varieties of English (scarcely represented in written texts) and various contact varieties (McConvell and Meakins, 2005, Sandefur, 1986, Shnukal, 1996). In addition to the language use of indigenous people, there has also been a huge change to the language picture of Australia as a result of migration in the last half century (Clyne 2005). Ideally, all of this diversity will be represented in the AusNC.

Initially, at least, this is unlikely to result in any new data collection. The intention is instead that the AusNC should have at least two major divisions.

One of these will be the carefully planned core component discussed in the previous section, while the second will have more of the nature of a text archive (See Peters, in press [2009] for discussion of this term). This component of the AusNC will be relatively unplanned and opportunistic in its accession of data, but the guiding aim will be to enable access to data about language in Australia in the widest sense. This will include, in addition to more standard varieties of English, indigenous languages, languages of migrant communities as used in Australia, indigenous varieties of English and contact varieties, varieties of English specific to different ethnic groups, and varieties of spoken English.

The audit of existing data which has begun will seek to identify holdings of any type of language data (English or other languages, text or multimodal) which is in a condition suitable for inclusion, or where the data can be brought to meet the technical standards of AusNC with a relatively small investment. In the future, researchers across all aspects of language in Australia will be encouraged to create data and metadata which meet the standards of AusNC so that such data can be added to the collection relatively easily.

3.3 Technical issues

The discussion of the preceding sections already implies that one crucial step in designing the AusNC is the creation and promulgation of a set of technical standards. These standards will have to specify the required formats of material which can be accepted into the corpus, the associated metadata which will be necessary for discovery, the discovery and access systems to be used, and a storage architecture (Cassidy, 2008).

One part of the Statement of Common Purpose from 2008 reads: "We further propose that such a corpus should be freely accessible and useful to the maximum number of interested parties", and this commitment leads naturally to a conception of the AusNC as a distributed group of resources meeting common standards which allow them to be linked by a set of network services. In most cases, users will interact with the corpus via a network connection (cf. the Corpus of Contemporary American English which is only available online, Davies, 2009).

Two crucial pieces in ensuring that such an architecture is possible will be well-understood metadata standards and a coherent approach to annotation. Metadata for linguistics resources has received a good deal of attention over the last decade (e.g. Bird and Simons, 2003). There are currently two well-developed standards which can be used at least as a basis for new projects: the Open Language Archives Community metadata scheme, and the IMDI metadata scheme.[5]

In order to ensure that data from a diverse range of sources can be stored in a way which that makes that data maximally useable for as many people as possible the use of a design based on stand-off annotation (Ide & Suderman, 2007) is a crucial design principle for the AusNC. Treating annotation as distinct from primary data will ensure that data is multi-purpose and maximally accessible for diverse types of research. This approach will also have the advantage of making multimodal data tractable. The data to which stand-off annotation relates need not be text data; what is essential is that the annotation is precisely linked to some section of primary data. The primary data itself might be text or it might be a section of an audio recording specified by time codes, and the annotation can be a transcript of the specified section of a recording, just as tagging for parts of speech might be the annotation for a specified segment of text. The use of stand-off annotation makes the two possibilities conceptually equivalent.

4 Implications for Language Technology

One of the research communities which will be serviced by an AusNC is the language technology community. The purpose of this section is to sketch some of the areas in which the project may be expected to impact on research in language technology. An important component of this resource is that it be sufficiently similar to the BNC and the ANC so that meaningful comparative work can be undertaken. The AusNC will also aim to include good samples of recently emerging genres, including computer-mediated communication, an increasingly important dimension of any type of language research. Such data will be freely accessible with copyright and ethical issues settled in advance. In some cases, this may mean that some data will have access or usage restrictions imposed on it, but these will be clearly indicated in metadata records and provided as part of the discovery tools.

Firstly, and most obviously, an AusNC will provide an easily accessible source of language samples taken from Australian usage which can be used for testing hypotheses and tools. In developing more accurate speech recognition systems, for instance, an AusNC will hold spoken language data that has been annotated not only instrumentally, as traditionally undertaken in speech recognition science, but also for what is "hearable" in the sense of

[5] Open Language Archives Community:< http://www.language-archives.org/>; IMDI: < http://www.mpi.nl/IMDI/>

being interactionally meaningful according to language use researchers, in particular conversation analysts. A detailed comparison of these different approaches to the annotation of the same set of language data is likely to be mutually beneficial for both fields.

In human-computer interaction studies as well a large collection of annotated human-human interactions, and subsequent comparisons with newly developed human-computer interactional systems, will allow for the kinds of statistical analysis that are so important to the field (Dale, 2005), as well as enabling closer analysis of differences between human-human and human-computer communication (Viethen and Dale, 2009).

Current plans emphasize a dynamic structure for AusNC, with data being added to the collection over time. Ideally this will lead to a collection which can be used to answer questions about changes in language use across time. The static nature of the BNC is becoming a significant issue, as research based on that resource does not necessarily generalise to contemporary usage. Ongoing maintenance and expansion will be included as part of the corpus design for AusNC but any solution depends on the level of funding which is available for ongoing work, and this is not a variable whose value can be foreseen.

One particular use of the AusNC flowing from this component will be in localization research (see for example Shreve 2006). The availability of a large corpus of specifically Australian English will be of great value in, for example, establishing local usage in respect of terminology and in the detailed investigation of other conventions specific to Australian English. Although the available resources will be less extensive, the AusNC will also be of use where localization of other languages for an Australian audience is at issue.

The preceding paragraphs have discussed some of the ways in which an AusNC would provide access to relevant data for language technologists. But language technologists would also have an important role in developing the tools which would provide that access. Various aspects of the design discussed in section 3 pose interesting problems in this regard, especially the inclusion of large quantities of multimodal data. Access to such data via rich metadata is straightforward, but ultimately direct access to the media would be enormously desirable. Some steps in this direction are being taken (e.g. Gaudi: Google Audio Indexing, Alberti et al. 2009), but there is great potential for research in this area (Baker et al., 2009). In addition to the problems of discovery, there are also problems in delivering specified segments of audio or video to a web browser on demand. Again, this is an area in which some research has taken place, including in Australia (Annodex, Pfeiffer et al., 2003), but it is also an area with great scope for further work.

These last two examples illustrate a more general point. The development of the technical infrastructure of any project such as an AusNC will offer a wide variety of possibilities for language technology research. The design of metadata standards and of discovery and access software will all require a great deal of new research and much of this will crucially depend on work in language technology.

5 Conclusion

In this paper, we have described the current state of the AusNC initiative, the plans which have been made to date and the first steps which have been taken towards implementing those plans. The community of language technology researchers in Australia is a community which must have a considerable stake in any such project, and we have also tried to set out some of the areas in which the field of language technology could contribute to and benefit from a resource such as AusNC.

Perhaps the most important point to take from this paper is that, although some general principles are emerging, the design of an AusNC is still very much negotiable. Language technologists can and should make their needs and preferences known. Such input can influence the shape of any project which does finally eventuate and it is in the interests of everybody that any project should be designed to be as useful as possible to as many different research communities as possible.

Appendix

Members of the AusNC Steering Commmittee:

Associate Professor Linda Barwick (Sydney)
Professor Kate Burridge (Monash)
Associate Professor Steve Cassidy (Macquarie University)
Professor Michael Clyne (Monash/Melbourne)
Associate Professor Peter Collins (UNSW)
Professor Alan Dench (UWA)
Professor Cliff Goddard (UNE)
Dr Michael Haugh (Griffith)
Professor Bruce Moore (ANU)
Dr Simon Musgrave (Monash)
Professor Pam Peters (Macquarie)
Professor Roly Sussex (Queensland)
Dr Nick Thieberger (Melbourne/Hawai'i at Manoa)

Wiki at: https://sakai-vre.its.monash.edu.au/access/wiki/site/89e714f1-79dd-4f1c-b031-2591b9d0a9fb/home.html

References

Alberti, Christopher, Michiel Bacchiani, Ari Bezman, Ciprian Chelba, Anastassia Drofa, Hank Liao, Pedro Moreno, Ted Power, Arnaud Sahuguet, Maria Shugrina, Olivier Siohan. (2009). 'An audio indexing system for election video material.' In *Proceedings of 2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp.4873-4876).

Baker, Janet, Li Deng, James Glass, Sanjeev Khudanpur, Chin-Hui Lee, Nelson Morgan and Douglas O'Shaugnessy. (2009). Research developments and directions in speech recognition and understanding. Part 1. *IEEE Signal Processing Magazine* 26(3):75-80.

Biber, Douglas, Susan Conrad and Randi Reppen. (1998). *Corpus linguistics: Investigating language structure and use*. Cambridge: Cambridge University Press.

Bird, Steven and Gary Simons. (2003). Seven dimensions of portability for language documentation and description. *Language*, 79, 557-582.

Cassidy, S. (2008) Building infrastructure to support collaborative corpus research, paper presented at the HSCNet Workshop on Designing the Australian National Corpus, UNSW, 4-5 December 2008.

Clyne, Michael. (2005). *Australia's language potential*. Sydney: University of New South Wales Press.

Dale, Robert. (2005). Human communication from the perspective of natural language processing. Paper presented at ConCom05, Conceptualising Communication. University of New England, 8-9 December 2005.

Davies, Mark (2009) The 385+ Million Word Corpus of Contemporary American English (1990-2008+): Design, Architecture, and Linguistic Insights. *International Journal of Corpus Linguistics* 14: 159-90.

Dixon, R. M. W. (2002) *Australian Languages*. Cambridge UK: Cambridge University Press.

Haugh, M., K. Burridge, J. Mulder, and P. Peters (eds.). (In press [2009]). *Selected Proceedings of the 2008 HCSNet Workshop on Designing the Australian National Corpus: Mustering Language*. Somerville, MA: Cascadilla Proceedings Project.

Haugh, Michael. (In press [2009]). 'Designing a multimodal spoken component of the Australian National Corpus.' In Haugh et al. (eds).

Ide, Nancy. (In press [2009]). 'The American National Corpus: Then, now and tomorrow". In Haugh et al. (eds).

Ide, Nancy and Catherine Macleod. (2001). The American National Corpus: A Standardized Resource of American English. *Proceedings of Corpus Linguistics 2001*, Lancaster.

Ide, Nancy, and Keith Suderman. (2007). 'GrAF: A graph-based format for linguistic annotations'. In Boguraev, B., N. Ide, A.Meyers, S.Nariyama, M.Stede, J.Wiebe et al. (eds) *The LAW: Proceedings of the Linguistic Annotation Workshop* (pp. 1-8). Stroudburg PA: Association for Computational Linguistics.

Lampert, Andrew. (In press [2009]). 'Email in the Australian National Corpus'. In Haugh et al. (eds).

Leech, Geoffrey. (1992). 100 million words of English: The British National Corpus (BNC). *Language Research, 28*, 1-13. UK.

McConvell, Patrick and Felicity Meakins. (2005). Gurindji Kriol: A mixed language emerges from code-switching. *Australian Journal of Linguistics*, 25 (1), 9-30.

Musgrave, Simon and Sarah Cutfield. (In press [2009]) 'Language Documentation and an Australian National Corpus' In Haugh et al. (eds).

Nelson, Gerald (1996) "The design of the corpus". In S. Greenbaum (ed.) *Comparing English Worldwide: The International Corpus of English*, 27-35. Oxford: Clarendon Press.

Peters, Pam (In press [2009]) 'The Architecture of a Multipurpose Australian National Corpus'. In Haugh et al. (eds).

Pfeiffer, Silvia, Conrad Parker and Claudia Schremmer. 2003. Annodex: a simple architecture to enable hyperlinking, search & retrieval of time--continuous data on the Web. In *Proceedings of the 5th ACM SIGMM international Workshop on Multimedia information Retrieval* (pp. 87-93). Berkeley, California.

Sandefur, John R. (1986). *Kriol of North Australia: A language coming of Age*. Work Papers papers of SIL-AAB: Series A, Volume 10.

Shnukal, Anna. (1994). Torres Strait Creole. In Nick Thieberger & William McGregor (Eds.), *Macquarie Aboriginal words* (pp. 374-398). Sydney: The Macquarie Library Pty Ltd.

Shreve, Gregory M. (2006). 'Corpus enhancement and computer-assisted localization and translation'. In Keiran J. Dunne (ed.) *Perspectives on Localization* (pp.309-331). Amsterdam/Philapdelphia: John Benjamins.

Thieberger, Nick and Simon Musgrave. (2007). Documentary linguistics and ethical issues. In Peter K.Austin (ed.), *Documentary and Descriptive Linguistics, Vol. 4* (pp.26-37). London: School of Oriental and Asian Studies.

Viethen, Jette and Robert Dale. (2009). Referring expression generation: what can we learn from human data? In *Proceedings of the Pre-Cogsci Workshop on Production of Referring Expressions: Bridging the Gap between Computational and Empirical Approaches to Reference*, 29 July 2009, Amsterdam, The Netherlands.

Corpus-based Extraction of Japanese Compound Verbs

James Breen and **Timothy Baldwin**
Department of Computer Science & Software Engineering
University of Melbourne
Australia
jimbreen@gmail.com, tb@ldwin.net

Abstract

We describe two methods for Japanese compound verb (JCV) extraction, based on synthesis and pattern matching over the Google Japanese n-gram corpus. We devise a number of filters to boost the precision of the corpus-based method, and evaluate the two methods based on a sample of JCVs occurring in varying frequency bands. We also investigate the distribution of JCV token frequency, and the type frequency of their components.

1 Introduction

This paper describes work conducted in a project to extract Japanese compound verbs (JCVs) from corpora and corpus-based resources. Compound verbs in Japanese have attracted considerable attention in Japanese linguistics as they are a highly productive and flexible element of the language (Shibatani, 1990; Baldwin and Bond, 2002; Tsujimura, 2006). Apart from some manually-prepared verb lists they have received relatively little attention in corpus linguistics.

The reasons for collecting and studying Japanese compound verbs include:

a. the development of reliable methods for extraction of the verbs from corpora;

b. investigation of the distribution of the verbs and their constituents;

c. investigation of the coverage of the verbs in the major lexicons

In particular, it is hoped that by isolating JCVs which are in use, but are not currently recorded or lexicalized, and eventually by developing and verifying Japanese meanings and English translational equivalents for these verbs, the lexicon of JCVs can be expanded.

At the current stage of the project, two methods for extracting compound verbs have been developed and applied over a major Japanese corpus. The result has been the identification of a large number of potential JCVs, which we show to have a high level of precision, relative to a sample of JCVs across varying frequency bands. We also investigate the distribution of the frequency of the verbs and their components.

2 Overview of Japanese Compound Verbs

The compound verb in Japanese (複合動詞 *fukugôdôshi*, hereafter JCV) is a concatenation of two or more verbs which function as a single multiword verb. There are several classes of JCV, however in this work we concentrate of the largest and most common class in which the first verb is in the continuative form (also known as the *masu*-stem because it forms the base for the polite spoken *-masu* group of inflections) (Uchiyama et al., 2005; Kubota, 1992).[1] In common with most studies of JCVs, we concentrate on verbs where both components are native Japanese verbs, not loanwords or Sino-Japanese words. This exclusion is because these latter verbs are much less

[1] In some cases this undergoes phonetic variation, e.g. the gemination in 引っ越す *hikkosu* as an alternative to the regular 引き越す *hikikosu*.

common and have a different morphology.

As a JCV consists of two adjacent verb components, we will refer to these components as the V1 and V2. A typical JCV is 行き過ぎる *ikisugiru* "to go too far", where the V1 is the continuative form of 行く *iku* "to go" and the V2 is 過ぎる *sugiru* "to be excessive; to be too much". 過ぎる is a particularly productive V2. A less productive V2 is 汚す *yogosu* "to make dirty", found in the JCV 食べ汚す *tabeyogosu* "to eat messily".

JCVs play a role in Japanese which is analogous to several different structures in other languages. English equivalents include compound verbs (e.g. *to start to eat*), verb-plus-gerund (e.g. *to start swimming*) and verb particle constructions (e.g. *to kick up (ball, fuss), to pull down*).

The JCV is a highly productive form, with some particular V1s and V2s being strongly represented, however there is no real restriction on a verb being used within a JCV, subject to issues such as aspect and valency (Kubota, 1992), and the result being meaningful. Some popular references list many hundreds of JCVs (Tagashira and Hoff, 1986) and major dictionaries typically include several thousand as entries, however it is generally recognized that many more JCVs are in use than are lexicalized. The incompleteness of the lexicalization of JCVs arises not only from their productivity, but from the fact that their meaning is often obvious to a Japanese speaker, and hence dictionary editors usually concentrate on JCVs which are polysemous, or have idiosyncratic meanings. Extension of the coverage of recorded and translated JCVs would be of assistance in areas such as language learning, and in lexicons used by morphological analysis and machine translation systems.

An example of a polysemous JCV is 引き抜く *hikinuku*, from 引く *hiku* "to draw; to pull", and 抜く *nuku* "to extract". It means both "to uproot" and "to pull out", and less obviously "to headhunt" and "to lure away".

3 General Approach and Resources

3.1 Approaches

A fundamental problem with searching Japanese corpora for unrecorded words is that Japanese text does not usually have spaces or any other marking between words. Thus the identification of words in text necessitates the use of a morphological analysis process to separate the words, and all such processes currently rely on extensive lexicons. The absence of a word in the lexicon usually results in the analysis software defaulting to producing a sequence of untagged morphemes until it can resynchronize.

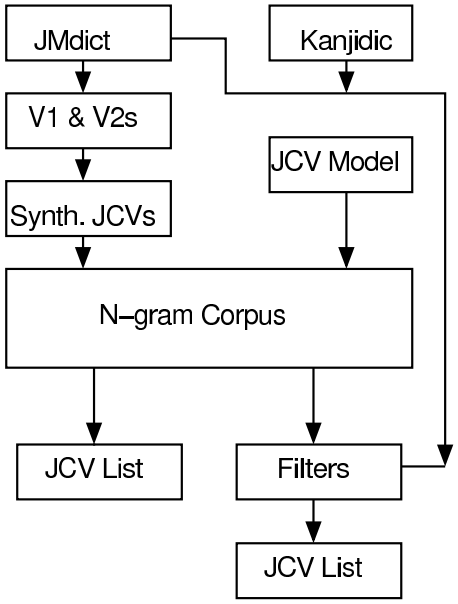

Figure 1: An outline of the two proposed approaches for JCV generation/extraction

An approach that has had some previous success is to synthesize possible words by mimicking Japanese morphological processes, and then testing, e.g. using a WWW search engine, to determine whether the word is in use (Breen, 2004a). A variant of this approach has been applied in the current project. A second approach in which the Google n-gram corpus (described below) was scanned using a filter designed to detect the character patterns consistent with JCVs. These approaches are described in detail below. Figure 1 shows a diagram of the two approaches.

3.2 Resources Used

The project uses several lexical resources to assist with the identification and extraction of JCVs. The JMdict Japanese-English dictionary database (Breen, 2004b) and the associated Kanjidic database (Breen, 2009) were used to establish sets of possible V1 and V2 components, and

a combined lexicon was constructed from the JMdict file and the following:

- the Kôjien Japanese dictionary (Ootsuka, 1998)
- the Daijirin Japanese dictionary (Matsumura, 1995)
- the GoiTaikei lexicon (Ikehara et al., 1997)
- the Japanese Linguistics Database (JLD) (Halpern, 2008)

The Kôjien and Daijirin are major Japanese dictionaries with extensive coverage of the Japanese lexicon. The GoiTaikei and JLD commercial lexicons are primarily used for Japanese NLP projects and research. The combined lexicon contains over 650,000 surface forms of Japanese words.[2]

The main corpus used in this project has been the Google Japanese n-grams (Kudo and Kazawa, 2007). The set of n-grams in the corpus was compiled by extracting text from a complete crawl of Japanese WWW pages for the month of July, 2007, and analyzing that text using the MeCab morphological analysis system (Kudo, 2008). n-gram sequences of up to 7-grams are recorded in the corpus if they were identified in 20 or more text segments. As systems such as MeCab are observed to break JCVs into up to three morphemes depending on the inflection of the verb, only the 1-gram, 2-gram and 3-gram section of the corpus were used in this study, and then only those n-grams which began with a *kanji* character.

A similar Google n-gram corpus has been used successfully in the extraction of verb-particle constrictions in English (Kummerfeld and Curran, 2008).

[2]Japanese is written using a combination of Chinese characters (*kanji*) and two syllabaries: *hiragana* and *katakana*. It has considerable flexibility as to whether words are written in *kanji*, one of the syllabaries, or a mixture. Also, alternative *kanji* are often used. For example, the JCV 詰め合わせる *tsume-awaseru* "to pack an assortment of goods, etc." can also be written: 詰め合せる, 詰合わせる, 詰合せる, 詰めあわせる or つめあわせる, and 並び変える *narabikaeru* "to put things in order" can also be written 並変える, 並び替える, 並替える, etc..

3.3 Synthesis of Compound Verbs

In this approach, a set of JCVs were synthesized as follows:

a. The JMdict lexicon was examined and JCVs identified. Including alternative surface forms, some 2,900 JCVs in which *kanji* were used in both the V1 and v2 were extracted. These were divided into the V1 and V2 components, yielding approximately 700 V1s and 600 V2s.

b. Using the V1 and V2 components, 420,000 synthetic JCVs were created via all combinations of the V1s and V2s. For each verb two forms were generated: the form in which the V2 used *kanji* as the root of the verb, and the form in which the V2 was entirely in the *hiragana* script. Both these forms are freely used in Japanese, for example 抱き付く *dakitsuku* "to cling to; to embrace" can equally well be written 抱きつく, and in fact the latter is more commonly used.

   For each of these, as well as the plain non-past tense (which is considered to be the reference form of Japanese verbs and is used for dictionary headwords) two inflections were generated: the continuative *te*-form and the plain past tense. These three are the most commonly used inflections in written Japanese, and it was considered appropriate to focus on them in order to detect whether words were in use.

Each synthetic JCV was initially checked against the combined lexicon, resulting in a total of 6,094 matches.

Each synthetic JCV was then checked against the Google n-gram corpus. As there were three inflections of two written forms of 420,000 JCVs, a total of 2,520,000 words were tested. The sections of the n-gram files which began with a *kanji* were preprocessed to recombine each 2-gram and 3-gram into a single character string, then sorted, resulting in a file of 270M unigrams in a file of 5.8GB. This facilitated processing in a single pass against the sorted verb file, thus enabling a rapid comparison and collation of results.

Initially, approximately 26,000 of the synthesized JCVs were matched in one or more of their

inflections. On inspection, the JCV form in which the V2 was in *hiragana* did not contribute significantly to the matches, and as this form has an increased chance of homophones which cannot be resolved without textual context, it was removed from the analysis. Also removed were a number of JCVs which were effectively alternative conjugations (passive, potential, etc.). This reduced the matched JCVs to 22,692.

Of the 6,094 JCVs which were found in the combined lexicon, 4,779 matched n-grams in the corpus, i.e. 1,315 which were found in the combined lexicon were not in the corpus. On inspection it was noted that many of the 1,315 were archaic and literary words.

The distribution of the counts of occurrences in the corpus is sharply asymptotic, with a small number having very high counts and declining to a long tail with over 15,000 having counts below 500.

3.4 Direct n-gram Search

In addition to the synthesis approach, an alternative approach was devised in which the n-gram corpus was scanned for character strings which conformed to the structural pattern of JCVs. From the examination of known JCVs, it can be determined that the common structural pattern is:

- a V1 consisting of one or two *kanji* followed by one to three *hiragana*;
- a V2 consisting of one or two *kanji* followed by one to four *hiragana*.

As there are many other valid text fragments which also conform to this pattern, e.g. noun/particle/verb, noun/particle/adjective, etc. filters were applied as follows:

a. the V1 component was limited to the *masu*-stems of known or potential verbs. To do this, a list of verb *masu*-stems was created and merged from:
   i. all the verbs in the JMdict dictionary;
   ii. all the V1 components used in the synthesis methods;
   iii. all the kanji in the Kanjidic database which had the potential to form a verb (this information is detailed in the database.)

   A total of 6,023 actual or potential verb stems were identified and used to filter the potential JCVs.

b. the inflecting part of the V2 was limited to the *hiragana* strings associated with valid verbs in the plain non-past, plain past and *te*-form inflections. A list of 208 such inflections was compiled and used as a filter.

A scan of the n-gram corpus for unigrams which conformed to the structural model and passed the filters yielded just on 135,000 potential JCVs. As these included many inflected forms, a considerable amount of post-processing was carried out to reduce them to a consolidated set of reference (plain non-past) forms. Some of the processes involved included:

a. matching the inflected and reference forms and combining the counts;

b. detecting and removing additional inflections such as the potential and passive forms, which share some of the inflection patterns of the reference form;

c. detecting and removing adjectives. In Japanese, adjectives inflect in a manner similar to verbs, and a number had been collected in the scan.

From this a reduced list of approximately 80,000 potential JCVs was produced. When tested against the combined lexicon, 6,203 matched with one or more of the dictionaries, an increase of 1,424 over the synthesis method. It is clear that this approach has an improved recall, i.e. the number of actual JCVs identified as a proportion to the number in existence, relative to the published lexicons, but possibly at the price of a reduced precision, i.e the proportion of actual JCVs among the potential JCVs. As with the synthesized JCVs, the distribution of the counts is asymptotic with a long tail.

4 Analysis of the Potential Compound Verbs

A detailed comparison of the potential JCVs compiled in the two approaches revealed that all the synthesized JCVs which had matched unigrams

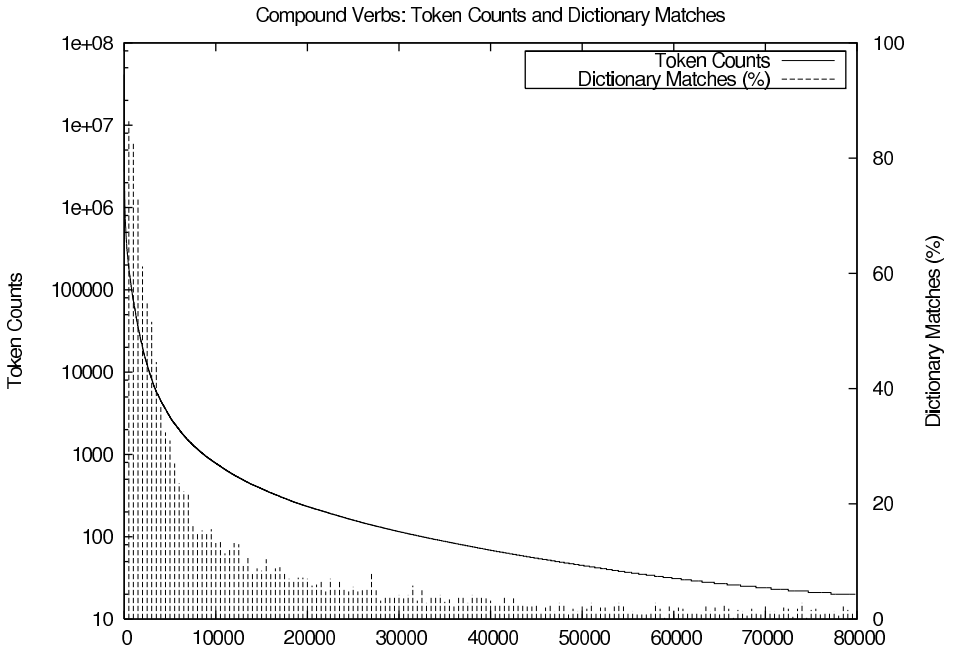

Figure 2: Analysis of the token counts and dictionary matches of JCVs, ranking in decreasing order of token frequency

in the corpus has also been collected in the search, and moreover the n-gram counts were almost always identical. This meant that a combined set could be used for further analysis, with tagging as to whether a JCV had been detected by both methods, or by the search alone.

Figure 2 shows the distribution of the n-gram counts and also the proportion of the potential JCVs which were found in a lexicon.[3]

A key issue is the extent to which these methods have revealed actual JCVs as opposed to character sequences which simply share symbolic characteristics with JCVs. To examine this aspect fully would require an evaluation of each JCV candidate in context, e.g. as it is used in WWW texts, to determine its status. In order to estimate the effectiveness of the JCV extraction approaches, samples of 50 potential JCVs were selected at random from each of three bands based on token frequency

a. High: JCVs with over 5,000 counts in the n-gram corpus (3,795 JCVs)

b. Medium: JCVs with 1,000 to 4,999 counts (4,886 JCVs)

c. Low: JCVs with 20 to 999 counts (71,138 JCVs)

The sample JCV candidates were classified as to whether they were in a lexicon or not, and if not, whether they were actually verbs. For the latter analysis, each potential JCV was manually checked against WWW pages via a search engine to verify whether it was being used as a verb. (At some later stage it may be possible to employ deeper linguistic analysis to carry out this process automatically.)

The summary of this classification is in Table 1. The figures in parentheses are numbers of JCVs in each category resulting from the search approach alone.

The JCV candidates which were classified as "other", i.e. not verbs, fell into several categories. The most common were inflected adjectives which had not been detected in filtering, adverbs such as 再び, V1s such as 見て which probably should have been filtered out (see below), and apparent typographical or grammatical errors associated with other verbs. Some were other constructs such as noun/verb without the usual intervening particle.

[3]For the purposes of depicting this, JCVs were examined in batches of 500, and the percentage which matched were plotted.

	High	Med	Low
In lexicon	27 (3)	12 (2)	1 (0)
Not in lexicon	23 (14)	38 (20)	49 (35)
verb	7 (1)	26 (8)	27 (14)
other	16 (13)	12 (12)	22 (21)

Table 1: Analysis of sample JCV candidates over the three frequency distribution bands, in terms of their occurrence in the lexicon; for JCV candidates not in the lexicon, we additionally break down the counts into verb and non-verb candidates

	High	Med	Low	Total
JMdict	1,788 (0.47)	453 (0.09)	671 (0.01)	2,912
Kôjien	1,420 (0.37)	401 (0.08)	976 (0.01)	2,797
Daijirin	1,626 (0.43)	491 (0.10)	970 (0.01)	3,087
GoiTaikei	1,375 (0.36)	377 (0.08)	661 (0.01)	2,413
JLD	2,172 (0.57)	932 (0.19)	2,023 (0.03)	5,126

Table 2: Occurrence of potential JCVs in the different dictionaries across the three frequency bands, in terms of the raw type count and proportion of overall types (in parentheses)

Of considerable interest is the relative performance of the JCVs identified by the synthesis approach; in each of the three bands almost all of these JCVs were valid verbs.

In terms of the precision of the different approaches, the full set of potential JCVs achieved precisions of 0.68, 0.76 and 0.56 respectively across the three selected bands, however within this the synthesis approach achieved precisions of 0.91, 1.00 and 0.93.

The comparative recall is difficult to measure as there is no gold standard for the number of JCVs in use or able to be used. Certainly the direct search approach achieved a greater recall but at the price of a lower precision.

As reported above, 6,203 of the potential JCVs matched with one or more of the dictionaries which make up the combined lexicon. It was noticed that while the high-ranking JCVs tended to match all the dictionaries, lower-ranking JCVs tended to match more sparsely, with often only one or two dictionaries matching. The specific dictionary matches were extracted for each of the high, medium and low bands. These are shown in Table 2. The figures in parentheses are the proportions of the dictionary matches against the total dictionary matches for the band.

	Combined n-gram	Synthetic n-gram	Lexicon	First 10,000
V1	2,601	680	1,294	1,290
V2	8,883	591	1,314	1,597

Table 3: V1 and V2 frequencies for the two proposed methods and in the lexicon

5 Productivity Measures of V1 and V2 Components

The productivity of the JCV is well known, as is the frequency with which some V1 and V2 components appear. It is useful, having established a reasonably large collection of JCVs, to use this to analyze the frequency of usage of the components.

For the purpose of this analysis, the V1s and V2s were extracted, counted and ranked from:

- the full collection of possible JCVs collected from the n-gram corpus;
- the synthesized JCVs which had matches in the n-gram corpus;
- the JCVs extracted from the combined lexicon;
- the highest-ranked 10,000 JCVs in the full collection (to see if there is a bias in component use according to the how common the JCV is).

In addition, V2 rankings collected by Kubota (1992) from her own corpus analysis and from an earlier published collection (Nomura and Ishii, 1987) were added for comparison.

The number of V1s and V2s which were extracted are shown in Table 3. As can be seen, over 80% of the V2s in the combined file are only found in the relatively low-frequency JCVs.

While there was some correlation of the frequency rankings of the V1s and V2s, there were also some notable differences. This can be seen in Tables 4 and 5, which show the 20 most common components as they were found in the combined JCV list, with their comparative rankings in the other lists.

With regard to the V1s in Table 4, it will be noted that there are some which do not appear in

V1	Combined n-gram	Synth. n-gram	Lexicon	First 10,000
通り	1	2	190	1
再び	2	1	642	3
思い	3	3	11	2
見て	4	4	520	4
限り	5	–	857	9
同じ	6	27	–	51
入り	7	13	52	29
余り	8	–	726	18
使い	9	11	45	13
出て	10	7	823	37
取り	11	5	1	5
入れ	12	19	49	24
行き	13	16	17	22
買い	14	24	24	11
持ち	15	22	25	19
作り	16	15	68	30
考え	17	17	113	25
言い	18	9	4	10
及び	19	–	1242	132
感じ	20	30	428	41

Table 4: V1 rankings for the two proposed methods, the lexicon and the most frequent 10,000 JCVs

V2	Combined n-gram	Synth. n-gram	Lexicon	First 10,000	Kubota
始める	1	1	6	1	1
続ける	2	2	3	2	8
過ぎる	3	4	4	5	9
出す	4	3	2	3	3
合う	5	5	5	6	4
得る	6	13	152	11	2
行く	7	10	59	12	–
初める	8	8	106	43	–
終わる	9	7	74	7	13
切る	10	11	9	10	5
難く	11	–	–	28	–
込む	12	6	1	4	7
多く	13	–	–	56	–
直す	14	9	10	9	11
見る	15	14	109	27	–
忘れる	16	15	18	16	39
上げる	17	12	8	8	21
歩く	18	19	50	25	34
来る	19	22	291	31	–
頑張る	20	–	–	68	–

Table 5: V2 rankings for the two proposed methods, the lexicon and the most frequent 10,000 JCVs

all lists, or have very different rankings. These almost all relate to differing interpretations at to what comprises a JCV. For example:

a. the V1s 限り *kagiri* "restricted", 余り *amari* "remain" and 及び *oyobi* "and", while deriving from verbs, are almost invariably use as conjunctions or adverbials in modern Japanese, and hence would not normally be part of a JCV. They should be added to the filter rules.

b. 再び *futatabi* "again" is more commonly regarded as an adverb, and should also probably be excluded.

c. the two "te-form" V1s (見て *mite* and 出て *dete*) could be classed as either part of JCVs, or as the common ($V1_{te-form}$,V2) sequence which has the sense of simultaneous occurrence or activity. They are often excluded from JCV classifications.

It will be noted that few of the high-ranking V1s in the lexicon appear in Table 4. On inspection it proved that they mostly lie in the 20–40 range in the n-gram lists. Given that dictionary compilers concentrate on JCVs which are polysemous or have idiosynchratic meaning, a lack of frequency alignment with corpus-based lists is to be expected.

While there is generally good agreement between the rankings of the lists of potential V2s in Table 5, some attract similar comments:

a. two of the potential V2s (難く *gataku* and 多く *ôku*) are clearly derived from adjectives, and should be added to the filter rules.

b. V2s which rank lower in the lexicon list (得る *eru* "to attain", 初める *hajimeru* "to start", etc.) are usually parts of semantically regular JCVs, and hence are less likely to be in a dictionary.

c. the appearance of 頑張る *gaNbaru* "to persist; to insist on; to stand firm" is of interest. It is not usually regarded as a JCV component, yet from its occurrence in the n-gram lists, it is clearly being used as such.

The foregoing comments are confined to the V1 and V2 components appearing among the 20 highest ranking counts in the combined list; similar comments can be made about a number of lower-ranking components. There is scope for considerable analysis of the ranking lists of V1 and V2 components.

6 Conclusions and Future Work

The work so far in the project has demonstrated that large numbers of JCVs are in regular use and can be detected through the application of NLP techniques to corpora. The two detection techniques which have been developed and tested have been demonstrated to have good levels of precision, especially in the case of the JCV synthesis method.

A substantial list of JCVs which are not recorded in commonly-used dictionaries has been identified for further study. In addition data on the frequency of usage of JCVs and their V1 and V2 components has been collected and can be made available for other Japanese NLP projects.

Future work in the project will include the development and testing of the meanings of unrecorded JCVs. The approach developed in earlier work (Uchiyama et al., 2005), which employs rule-based and statistical methods based on extensive classification of the V1 and V2 components, will be followed.

7 Acknowledgment

The assistance of Mr Jack Halpern, who has kindly provided a copy of his Japanese Linguistic Database for use in the project, is gratefully acknowledged.

References

Timothy Baldwin and Francis Bond. 2002. Multiword expressions: Some problems for Japanese NLP. In *Proceedings of the 8th Annual Meeting of the Association for Natural Language Processing (Japan)*, pages 379–382, Keihanna, Japan.

James Breen. 2004a. Expanding the Lexicon: the Search for Abbreviations. In *Proceedings of the Papillon Workshop*, Grenoble, France.

James Breen. 2004b. JMdict: a Japanese-Multilingual Dictionary. In *Proceedings of the COLING*

2004 Multilingual Linguistics Resources Workshop, pages 65–72, Geneva, Switzerland.

James Breen. 2009. Kanjidic/Kanjd212 Project. http://www.csse.monash.edu.au/~jwb/kanjidic.html.

Jack Halpern. 2008. Japanese Lexical Database. http://www.cjk.org/cjk/samples/japword.htm.

Satoru Ikehara, Masahiro Miyazaki, Akio Yokoo, Satoshi Shirai, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi. 1997. *Nihongo Goi Taikei – A Japanese Lexicon*. Iwanami Shoten. 5 volumes. (in Japanese).

Mariko Kubota. 1992. *Japanese Compound Verbs*. Ph.D. thesis, Monash University.

Taku Kudo and Hideto Kazawa. 2007. Japanese Web N-gram Corpus version 1. http://www.ldc.upenn.edu/Catalog/docs/LDC2009T08/.

Taku Kudo. 2008. MeCab: Yet Another Part-of-Speech and Morphological Analyzer. http://mecab.sourceforge.net/.

Jonathan Kummerfeld and James Curran. 2008. Classification of verb particle constructions with the google web1t corpus. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 55–63, Hobart, Australia.

Akira Matsumura, editor. 1995. *Daijirin*. Sanseido, 2nd edition.

Masaaki Nomura and Masahiko Ishii. 1987. 複合動詞資料集 *(Fukugôdôshi Shiryôshû - Compound Verb Collection)*. National Language Research Institute, Tokyo.

Nobukazu Ootsuka, editor. 1998. *Kôjien*. Iwanami Shoten, 5th edition.

Masayoshi Shibatani. 1990. *The Languages of Japan*. Cambridge University Press, Cambridge, UK.

Yoshiko Tagashira and Jean Hoff. 1986. *Handbook of Japanese Compound Verbs*. Hokuseido Press (Tokyo).

Natsuko Tsujimura. 2006. *An Introduction to Japanese Linguistics*. Blackwell, Oxford, UK, 2nd edition.

Kiyoko Uchiyama, Timothy Baldwin, and Shun Ishizaki. 2005. Disambiguating Japanese Compound Verbs. *Computer Speech and Language*, 19(4):497–512.

Double Double, Morphology and Trouble: Looking into Reduplication in Indonesian

Meladel Mistica, Avery Andrews, I Wayan Arka
The Australian National University
{meladel.mistica,avery.andrews,
wayan.arka}@anu.edu.au

Timothy Baldwin
The University of Melbourne
tb@ldwin.net

Abstract

This paper investigates reduplication in Indonesian. In particular, we focus on verb reduplication that has the agentive voice affix *meN*, exhibiting a homorganic nasal. We outline the recent changes we have made to the implementation of our Indonesian grammar, and the motivation for such changes.

There are two main issues that we deal with in our implementation: how we account for the morphophonemic facts relating to sound changes in the morpheme; and how we construct word formation (i.e. sublexical) rules in creating these derived words exhibiting reduplication.

1 Introduction

This study looks at full reduplication in Indonesian verbs, which is a morphological operation that involves the doubling of a lexical stem. In this paper, we step through the word formation process of reduplication involving agentive voice marking, including the morphophonemic changes and the morphosyntactic changes brought about by this construction. The reduplication investigated here is a productive morphological process; it is readily applied to many lexical stems in creating new words. Instead of having extra entries in the lexicon for reduplicated words, we aim to investigate the changes brought about by reduplication and encode them in a meaningful way to interpret, during parsing, these morphosyntactically complex, valance-changing, derived words.

This investigation sits within a larger Indonesian resource project that primarily aims to build an electronic grammar for Indonesian within the framework of *Lexical Functional Grammar* (LFG). Our project forms part of a group of researchers, PARGRAM[1] whose aim is to also produce wide-coverage grammars built on a collaboratively agreed upon set of grammatical features (Butt et al., 1999). In order to ensure comparability we use the same linguistic tools for implementation.[2]

One of the issues we address is how to adequately account for morphophonemic facts, as schematised in Examples (1), (2) and (3):

(1) [me*N*+tarik]^2
↔ me*N*+tarik+*hyphen*+me*N*+tarik
↔ *menarik-menarik*
"pulling (iteratively)"

(2) me*N*+[tarik]^2
↔ me*N*+tarik+*hyphen*+tarik
↔ *menarik-narik* (**menarik-tarik*)
"pulling quickly"

(3) me*N*+[tarik]^2
↔ tarik+me*N*+*hyphen*+tarik
↔ *tarik-menarik* (**narik-menarik*)
"pull at each other"

Here, *tarik* "pull" is the verb stem, *meN* is a verbal affix with a homorganic nasal (the function of which will be discussed in Section 2.1), ^2 is the notation we use for reduplication, and the square brackets [ ] are used to specify the scope of the reduplication.

[1] http://www2.parc.com/isl/groups/nltt/pargram/

[2] http://www2.parc.com/isl/groups/nltt/xle/ and http://www.stanford.edu/~laurik/fsmbook/home.html

Each of the examples consists of three lines: (a) a simplified representation of which words are reduplicated, (b) a breakdown of the components that make up the surface word, and (c) the surface word (in italics). Note that the first-line representation for (2) and (3) is identical, but the surface words differ on the basis of the order in which the reduplication and *meN* affixation are applied. Note also that, as is apparent in the gloss, (3) involves a different process to the other two examples, and yet all three are dealt with using the same reduplication strategy in our implementation. We return to discuss these and other issues in Section 3.

The morphological analyser is based on the system built by Pisceldo et al. (2008), whose implementation of reduplication follows closely that suggested for Malay by Beesley and Karttunen (2003). However, (3) is not dealt with by Beesley and Karttunen (2003), and the solution of Pisceldo et al. (2008) requires an overlay of corrections to account for the distinct argument structure of (3). This paper outlines a method for reorganising the morphological analyser to account for these facts in a manner which is more elegant and faithful to the data.

2 Reduplication in Indonesian

2.1 About Indonesian

Indonesian is a Western Austronesian language that has **voice marking**, which is realised as an affix on the verb that signals the thematic status of the subject (Musgrave, 2008). In Indonesian, the subject is the left-most NP in the clause. Below we see examples of AV (agentive voice),[3] PV (patient or passive voice) and UV (undergoer voice — bare stem).

(4) *[Amir] **mem**baca buku itu*
Amir AV+read book this
"Amir read the book"

(5) *[Buku itu] **di**baca oleh Amir*
book this PV+read by Amir
"The book was read by Amir"

(6) *[Temannya] dia pukul*
his.friend he/she UV.hit
"He hit his friend"

[3] In (4) the *mem-* "AV- AGENTIVE VOICE" is actually *me* plus a homorganic nasal

The marking on the verb indicates the semantic role of the subject, in square braces [ ] the agent in (4), and the theme and patient in (5) and (6).

2.2 Productive Reduplication

Indonesian has three types of reduplication: partial, imitative and full reduplication (Sneddon, 1996). We only consider full reduplication — or full repeat of the lexical stem — for this study because it is the only type of reduplication that is productive. We encode three kinds of full reduplication in the morphological analyser:

(7) REDUPLICATION OF STEM

duduk-duduk	*sakit-sakit*
sit-sit	sick-sick
"sit around"	"be periodically sick"

(8) REDUPLICATION OF STEM WITH AFFIXES

membunuh-bunuh	*bunuh-membunuh*
AV+hit-hit	AV+hit-hit
"hitting"	"hit each other"

(9) REDUPLICATION OF AFFIXED STEM

membeli-membeli
AV+buy-AV+buy
"buying"

Reduplication seems to perform a number of different operations. There is an aspectual operation, which affects how the action is performed over time. These examples are seen in (7) *sakit-sakit* and (8) *membunuh-bunuh*. These are comparable to the English progressive *-ing* in *He is kissing the vampire* versus *He kissed the vampire*, where the former depicts an event performed over time and the latter a punctual one.

However, this operation is not exactly equivalent to the English progressive, as seen below:

(10) *Saya memukul-memukul dia*
1.SG AV+hit-AV+hit 3.SG
"I am/was hitting him"/"I repeatedly hit him."

(11) *#Saya membunuh-membunuh dia*
1.SG AV+kill-AV+kill 3.SG
"#I was killing him"

(12) *Saya* *membunuh* *binatang*
1.SG AV+hit-AV+hit animal
"I killed an animal"/"I killed animals"

(13) *Saya* *membunuh-membunuh* *binatang*
1.SG AV+hit-AV+hit animal
"I killed animal after animal"/"#I was killing the animal"

As can be seen, this operation cannot apply to the verb *bunuh* "kill" in (11) to mean "killing". However if the object can be interpreted as plural then the action can be applied to the multiple objects as shown in (13). So there is this sense of either being able to distribute the action over time repeatedly or distribute/apply the action over different objects, when the semantics of the event does not allow the action to be repeated again and again, such as killing one animal.[4] The examples in (7) show more semantic variation on reduplication, such as an additional meaning of purposelessness for *duduk-duduk* "sit around".[5]

Another function of reduplication is the formation of reciprocals, as shown with *bunuh-membunuh* in (8). This verb formation is clearly not simply a case of reduplicating an affixed stem; there is a more involved process. We see that this kind of reduplication involves valence reduction: in (14) we have a subject and an object that's expressed in the sentence, but in (15) we only have a subject expressed, which encodes both the agent and patient.

(14) *Mereka* *membunuh* *dia.*
they AV+kill him/her
"They kill him/her."

(15) *Mereka* *bunuh-membunuh*
they kill-AV+kill
"They kill each other."

3 Tools to Construct the Word

This section outlines the process for building up the word. We look at at the tools that are used and the theoretical framework upon which the tools are built.

[4]The example in (11) can only be felicitously used if your victim was part of the army of the undead - FYI.

[5]These types of examples will not be discussed further here as they do not exhibit agentive voice marking.

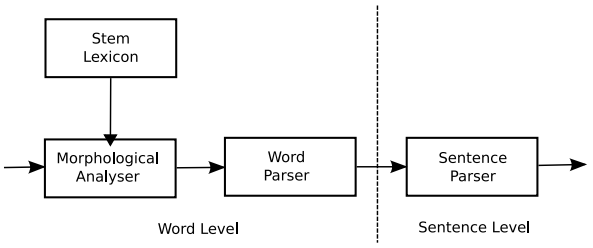

Figure 1: Pipeline showing word-level and sentence-level processes

Figure 1 is the overall course-grained architecture of the system. The dotted vertical line in Figure 1 delimits the boundary between sublexical processes and sentential (or partial) parsing. We are only interested in discussing the components to the left of this boundary, which is where the building of the word-level processes take place.

The components marked "Stem Lexicon" and "Morphological Analyser" utilise the finite state tools XFST and LEXC. The input to the morphological analyser is the sentence that has been tokenised, and its output is a representation of the words split into its morphemes. Furthermore, the first lines of each of the examples of (1), (2), and (3) seen earlier are the representation used, but simplified here to show only the required detail; they show the parts of the word are reduplicated and what other affixes are exhibited. This is then fed as input to the "Word Parser".

3.1 Theoretical Assumptions

The grammar formalism upon which the 'Word Parser' and 'Sentence Parser' are built is Lexical Functional Grammar (LFG). LFG has 'a parallel correspondence' architecture (Bresnan, 2001), which means relevant syntactic information is distributed among the parallel representations, and that the representations are related via mapping rules.

The level of representation that defines grammatical functions (subject, object etc.) and the constraints upon them, as well as features such as tense and aspect is called the **f-structure**. The f-structure is represented as attribute value matrices, where all required attributes must have unique and complete values. The **c-structure** is represented with phrase

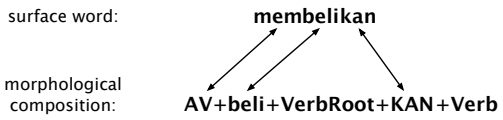

Figure 2: Upper and lower language correspondence for *membelikan* "buy someone something"

structure trees and describes the language-specific arrangement of phrases and clauses for a given language. This level of representation accounts for the surface realisation of sentences, such as word order. The **a-structure** specifies the arity of the predicate, defining its arguments and their relative semantic prominence, which have mapping correspondences to grammatical functions.

3.2 Finite State Tools: XFST and LEXC

The 'Morphological Analyser' is built with tools that provide access to finite-state calculus algorithms, in particular the XEROX FINITE-STATE CALCULUS implementation (Beesley and Karttunen, 2003). The finite-state network we create with these tools is a transducer, which allows for a lower language — or a definition of the allowable surface words in the language — and an upper language, which defines the linear representation of the morphological units in the surface word. An example of an upper language 'output', for analysis, and its corresponding lower language 'input' or morphological analysis is given in Figure 2.

In this example the *mem-* prefix is represented with AV+, the stem *beli* "buy" gets extra information about its part-of-speech via the +VerbRoot suffix, and the applicative *-kan* is represented as +KAN.

We encode the morphotactics of the Indonesian word with the XFST tool, which provides an interface to these algorithms for defining and manipulating the finite state networks, as well as LEXC, which is used for defining the lexicon (Beesley and Karttunen, 2003).

The Pisceldo et al. (2008) system, on which our system is based, employs the same finite state tools as the current implementation. It has two major components which are labelled **morphotactic rules** and **morphophenemic rules**. Figure 3 shows the general schema of the Pisceldo et al. (2008) system.

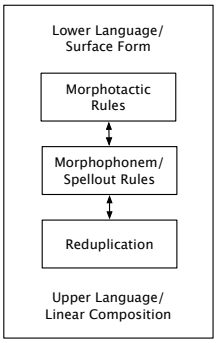

Figure 3: Pisceldo et al. (2008) morphological analyser

The label **reduplication** is a little misleading because it simply indicates when the doubling of the morphological form takes place. In XFST this process is named **compile-replace**. The compile-replace algorithm was developed to account for non-concatenative morphological processes, such as the vocalisation patterns in Arabic and full reduplication in Malay (Beesley and Karttunen, 2003). The compile-replace algorithm for reduplication works by delimiting the portion of the network that is affected by compile-replace. This so-called 'portion' of the net is defined as a regular expression and is delimited by the tags '^[' and '^]' on the lower side of the net and 'Redup[' and ']Redup' on the upper side. When the compile-replace algorithm is invoked, the net defined by regular expression between '^[' and '^]' is copied. There are computational limitations to what can be defined within these delimited tags, so in practice we apply compile replace to predefined lexemes, or stems, as listed in the LEXC stem lexicon, with optional predefined affixes, and exclude unknown stems.

3.3 Word Level Parser: XLE

The tool used for parsing, XLE, only utilises two of the three levels of representations discussed earlier: f-structure and c-structure. In Figure 1 both the 'Word Parser' and 'Sentence Parser' utilise XLE.

XLE is a grammar development environment which interprets grammars written in an electronic parseable variation of LFG. It is the tool used for defining the phrase structure, as well as the sublexical rules, which describes how the word is com-

posed. We construct these rules via c-structure rules, which look like traditional grammar rewrite rules but with annotations giving us the information that can only be encoded via the phrase structure. Within the "Word Parser" component, there are defined **sublexical rules** that are interpreted using XLE.

This component crucially relies on the analysis of the 'Morphological Analyser' and its output must be a meaningful representation of the input, which is the surface form of the reduplicated verb. There is a semantic motivation for wanting to represent the predicates in (1) *menarik-menarik*, (2) *menarik-narik*, and (3) *tarik-menarik* in different ways. We would want our morphological analysis to be sensitive to their semantic differences, however small or large. For these given predicates, there are three important components of the word to represent:

- reduplication: Redup[ ]Redup
- the agentive voice affix: AV
- the verb stem: *tarik* "pull"

We could represent the analysis of *menarik-menarik* as Redup[AV+tarik]Redup, but we would want to differentiate *menarik-narik* from this and so could represent this as AV+Redup[tarik]Redup. However, this also seems a plausible output for *tarik-menarik*, as does the former. In order to enforce a unique representation for all three, we arrive at:

(16) menarik-menarik: Redup[AV+tarik]Redup

(17) menarik-narik: AV+Redup[tarik]Redup

(18) tarik-menarik: Redup[tarikAV+]Redup

The first reduplicated example, *menarik-menarik* in (16), with the stem *tarik* "pull" means "pull again and again". The second example, *menarik-narik*, has a very similar meaning to (16), but the major difference is that the action (i.e. the "pulling" in the case of *tarik*) is repeated faster. The last example *tarik-menarik* , (18), means "pull at each other", in a tug-of-war fashion.

4 Integration into the Grammar

4.1 Reciprocals

From a formal point of view, it seems that the reciprocal is formed by marking two verbs with undergoer and agentive voice, which forms a linking between the agent and the patient of the action. In Indonesian, undergoer voice is the unmarked bare verb as shown by Arka and Manning (2008), and agentive voice is marked with *meN*. This compound verb analysis gives us an adequate semantic account of reciprocals, but more needs to be done in order to explain the arity reduction of the resulting predicate, as seen in (19) where *mereka* "they" is the only argument of the verb.

(19) *Mereka* *pukul-memukul*
they UV.hit-AV+hit
"They hit each other"

We adopt a similar analysis of reciprocals in Indonesian to the analysis of Alsina (1997) and Butt (1997) for causative verbs in Chichewa and permissives in Urdu, respectively: the reciprocal verb formation in Indonesian is a type of complex predicate in that the elements of the reciprocal combine to alter the argument structure of the resulting predicate, which acts as a single grammatical unit (Alsina et al., 1997). Even though the same principle of predicate composition applies, these analyses do not involve valence reduction as it does in Indonesian, but rather valence increasing.

Although the undergoer plus agentive voice treatment of reciprocal formation gives us a neat account of argument linking, these verb stems would then be considered two separate verbs as they both have their own voice marking, and therefore have their own values for the VOICE attribute in their f-structure attribute value matrices. This means, from an implementation point of view, there would have to be a semantic identity check to ensure both verbs have the same verb stem. For this implementation reason, we choose to keep this as a process within the 'Morphological Analyser' and as reduplication rather than verb compounding. This then saves a form of 'identity matching' of the two stems at a later stage.

The reciprocal is interpreted as such by virtue of the reduplication construction where the agentive voice affix *meN* is inserted between the reduplicated stems. Therefore the 'instructions', if you will, for composing reciprocals are encoded in the sublexical c-structure rules and manifested in the f-structure, as it affects argument linking.

If we step back from the implementation for a moment, we can represent schematically what happens to the arguments of a regular transitive verb such as (20), when it is composed as a reciprocal (21). But what we want is to create a general rule that allows this operation to apply to all transitive verbs where the resulting reduplicated form has an interpretable reciprocal predicate.

(20) *pukul* < agent, patient >

(21) *pukul-memukul* < agent&patient >

The important components of the reciprocal word forming sublexical rules are as follows:

- The input to the rule has one argument (ARG), which is a transitive stem verb that requires a subject (ARG SUBJ) and an object (ARG OBJ)
- The resulting complex predicate (RECIP-rocal) only requires a subject (SUBJ) that must be plural (NUM pl)

The input predicate ARG must still be complete, meaning that is must still satisfy its (ARG SUBJ) and (ARG OBJ), which is the **agent** and **patient** in (20). That is, the verb on which the RECIProcal verb is formed is transitive and requires all its arguments to be filled. We can achieve this via coindexing the subject and object of the input predicate ARG with the subject of the derived predicate RECIP.

(22) RECIP< (SUBJ$_i$), ARG < (SUBJ$_i$), (OBJ$_i$)> >

The resulting predicate is mono-valent, in that it only needs to satisfy a subject, however it has an input predicate. Figure 4 shows the resulting f-structure for the reciprocal sentence in (19). The first line (labelled PRED) is the representation of the semantics of the head of the attribute value matrix over which it has scope. In this case the PRED on the first line represents the main verb *pukul-memukul* "hit-reciprocally". It tells us it is a derived reciprocal whose first slot is satisfied by the attribute value matrix labelled 4, which is the subject; the second slot is satisfied by a verb that takes two arguments.

The c-structure for (19) is shown in Figure 5. Each of the numbered nodes corresponds to a component in the f-structure. It is clear in the c-structure

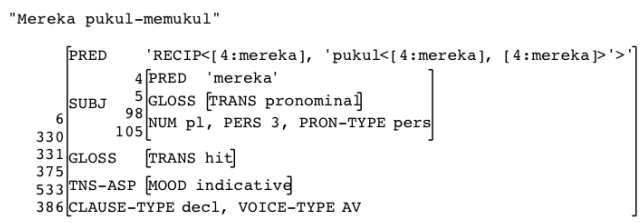

Figure 4: Feature structure

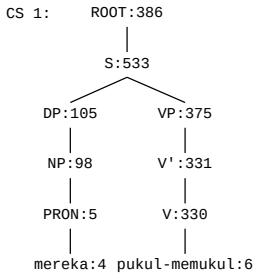

Figure 5: Constituent structure

that the verb only takes one noun phrase argument, which is the subject. The operation that composes the derived reciprocal verb requires a transitive verb as input, which is *pukul* "hit" in Figure 4, and it is represented in the f-structure inside the PRED value for the RECIP verb.

4.2 Distributed Reduplication

The implementation of the non-reciprocal reduplication is less involved, in that this construction simply triggers an additional feature in the f-structure, however it has its complexities too. The main issue is: what feature should be added?

We discussed earlier that reduplication constructions such as (23) are not exactly the same as the English progressive aspect, and in some examples have more of an iterative aspect, in that the action is repeated but not necessarily with one sustained action over time, but in a start-stop fashion. Therefore a feature such as ITER + as part of the tense-aspect definition of the clause could be added to the f-structure.

Noun phrases in Indonesian are underspecified for number, much like the English noun phrases that are headed with mass nouns, such as *rice*. However the

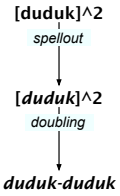

Figure 6: Spellout then doubling of *duduk*

Figure 7: Examples where spellout must precede doubling

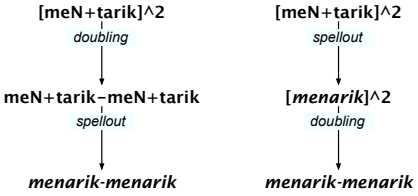

Figure 8: Examples where the order of double and spellout has no consequence

reduplication on the verb can impose a plural reading on the argument(s) of the verb, where the action is applied to each and every member of the argument of the verb, as seen in the second translation in (23) ((12) is an earlier example).

(23) *Dia* *memukul-mukul* *temannya*
He AV+hit-hit his.friend
"He was (repeatedly) hitting his friend."/"He hit each of his friends."

When the verb determines the number of its arguments, this is called a **pluractional** verb (Corbett, 2000). Pluractionality specifies that the action is over multiple affected objects, and so we could add the attribute-value pair PLURACT + for these constructions , which would not be part of the tense-aspect definition of the clause.

In the present implementation, for sentences such as (23), both solutions are possible.

5 Rejigging the Morphological Analyser

Traditional analyses of reduplication have been modelled on a theory of phonological copying or a doubling of a phonologically-rendered form. This entails that we begin with a lexeme *duduk* "sit", we then execute the spellout rule or the phonological rendering giving us *duduk* /dudu?/, and then this form is doubled producing *duduk-duduk* "sit around", as seen in Figure 6.

The architecture of the Pisceldo et al. (2008) morphological analyser in Figure 3 models this idea of how the reduplication mechanism works. Specifically, the morphophonemic rules are executed first, giving us our spelled-out rendering, which is then doubled. Certainly when we examine some of the morphophonemic facts of reduplication in Indonesian, it gives support for this architecture. Such an example is shown in Figure 7, which is the realisation of AV+[tarik]^2 (agentive voice prefix with the reduplicated stem *tarik* "pull"); Figure 8 presents a case where relative ordering does not matter.

However, this implementation cannot account for the full morphophemic facts of reduplication, namely the reciprocal construction, without the aid of corrective spellout rules.

We see in Figure 9 that for these types of examples we need to allow for the doubling of the verbs stem, ensuring appropriate attachment of voice marking to the respective stems, before we allow for spellout to take place. The notation (-,AV) is an indication of how the voice affixes are 'multiplied out' upon reduplication.

Inkelas and Zoll (2005) puts forward a theory of reduplication, Morphological Doubling Theory (MDT), that can incorporate both strategies allowing spellout and doubling in any order, and that both strategies are called for. They also claim that the reduplicated stems are a lot more discrete and can bear different affixes, and their phonological rendering can be realised independently from each other. This seems to model what we observe in the reciprocal construction in Indonesian: an independence of phonological realisation. The two different ordering for spellout and doubling very neatly separates

Figure 9: Examples where Doubling must precede spellout

out the two types of reduplication processes. Therefore within both the morphological analyser and the sublexical component, reciprocal reduplication and distributive reduplication are handled aptly as distinct separate processes, as seen in Figure 10.

Although we do not in whole borrow from MDT, some of the concepts put forward in the theory gave us cause to see the two reciprocal processes as being separate in the morphological analyser. As such, we have allowed for both spellout before reduplication and then spelling out this doubling process. We see these two processes as serving different purposes: one for the aspectual/distributed reduplication and the other for the reciprocal reduplication. It seems apt to be treating them differently in the morphological analyser, given that they are implemented so differently in the sublexical word building component.

6 Conclusion

In this study, we discussed reduplication in combination with the voice marker AV. There are other voice prefixes such as the passives *di*, *ter* and *ber* that we still need to investigate. We would want to see whether these would require special treatment. In addition, we need to investigate more deeply the interaction with applicative morphology such as *-kan* and *-i*, as shown in (24), and to ensure that we develop an analysis that would complement our existing implementation of the applicatives (Arka et al., 2009).

(24) *Mereka beli-membilikan mobil*
they AV+beli-beli+KAN car
"They bought cars for each other"

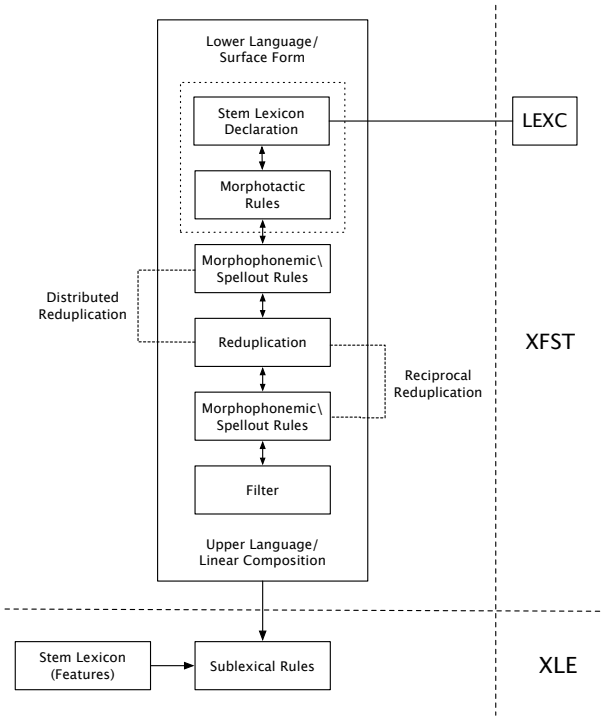

Figure 10: Current morphological analyser with separated doubling process for the two types reduplication constructions.

We had initially considered all reduplication in the morphological analyser as the same doubling process, and implemented reduplication accordingly. Although the two forms of reduplication we were investigating, reciprocal and distributional, were morphosyntactically very different and so had to be implemented very differently in the sublexical component, we had not considered handling them differently from each other in the morphological analyser to account for their differences with respect to their morphophonemic facts. Instead of preemptive corrective rules, we implemented another component to correctly treat the stems of the reciprocal reduplication and distributive reduplication as being more independent of each other, with respect to their phonological realisation.

References

Alex Alsina, Joan Bresnan, and Peter Sells. 1997. Complex predicates: Structure and theory. In Alex Alsina,

Joan Bresnan, and Peter Sells, editors, *Complex Predicates*, pages 1–12. CSLI, Stanford, USA.

Alex Alsina. 1997. Causatives in Bantu and Romance. In Alex Alsina, Joan Bresnan, and Peter Sells, editors, *Complex Predicates*, pages 1–12. CSLI, Stanford, USA.

I Wayan Arka and Christopher Manning. 2008. Voice and grammatical relations in Indonesian: a new perspective. In Simon Musgrave and Peter Austin, editors, *Voice and grammatical relations in Austronesian languages*, pages 45–69. CSLI, Stanford, USA.

I Wayan Arka, Avery Andrews, Mary Dalrymple, Meladel Mistica, and Jane Simpson. 2009. A computational morphosyntactic analysis for the applicative -i in indonesian. In *Proceedings of LFG2009*.

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications.

Joan Bresnan. 2001. *Lexical Functional Syntax*. Blackwell, Massachusetts, USA.

Miram Butt, Tracy Holloway King, Maria-Eugenia Nino, and Frederique Segond. 1999. *A Grammar Writers Cookbook*. CSLI, Stanford, USA.

Miriam Butt. 1997. Complex predicates in Urdu. In Alex Alsina, Joan Bresnan, and Peter Sells, editors, *Complex Predicates*, pages 1–12. CSLI, Stanford, USA.

Greville Corbett. 2000. *Number*. Cambridge University Pressl, Cambridge, UK.

Sharon Inkelas and Cheryl Zoll. 2005. *Reduplication: Doubling in Morphology*. Cambridge Studies in Linguistics, 106. Cambridge University Press, Dunno, USA.

Simon Musgrave. 2008. Introduction: Voice and grammatical relations in austronesian languages. In Simon Musgrave and Peter Austin, editors, *Voice and grammatical relations in Austronesian languages*, pages 1–21. CSLI, Stanford, USA.

Femphy Pisceldo, Rahmad Mahendra, Ruli Manurun, and I Wayan Arka. 2008. A two-level morphological analyser for Indonesian. In *Proceedings of the Australasian Language Technology Association Workshop*, volume 6, pages 88–96.

James Neil Sneddon. 1996. *Indonesian reference grammar*. Allen Unwin, St. Leonards, N.S.W.

Contrastive Analysis and Native Language Identification

Sze-Meng Jojo Wong
Centre for Language Technology
Macquarie University
Sydney, NSW, Australia
szewong@science.mq.edu.au

Mark Dras
Centre for Language Technology
Macquarie University
Sydney, NSW, Australia
madras@science.mq.edu.au

Abstract

Attempts to profile authors based on their characteristics, including native language, have drawn attention in recent years, via several approaches using machine learning with simple features. In this paper we investigate the potential usefulness to this task of contrastive analysis from second language acquistion research, which postulates that the (syntactic) errors in a text are influenced by an author's native language. We explore this, first, by conducting an analysis of three syntactic error types, through hypothesis testing and machine learning; and second, through adding in these errors as features to the replication of a previous machine learning approach. This preliminary study provides some support for the use of this kind of syntactic errors as a clue to identifying the native language of an author.

1 Introduction

There is a range of work that attempts to infer, from some textual data, characteristics of the text's author. This is often described by the term *authorship profiling*, and may be concerned with determining an author's gender, age, or some other attributes. This information is often of interest to, for example, governments or marketing departments; the application that motivates the current work is profiling of phishing texts, texts that are designed to deceive a user into giving away confidential details (Fette et al., 2007; Zheng et al., 2003).

The particular characteristic of interest in this paper is the native language of an author, where this is not the language that the text is written in. There has been only a relatively small amount of other research investigating this question, notably Koppel et al. (2005), Tsur and Rappoport (2007), Estival et al. (2007), and van Halteren (2008). In general these tackle the problem as a text classification task using machine learning, with features over characters, words, parts of speech, and document structures. Koppel et al. (2005) also suggest syntactic features, although they do not use them in that work.

The goal of this paper is to make a preliminary investigation into the use of syntactic errors in native language identification. The research drawn on for this work comes from the field of *contrastive analysis* in second language acquisition (SLA). According to the contrastive analysis hypothesis formulated by Lado (1957), difficulties in acquiring a new (second) language are derived from the differences between the new language and the native (first) language of a language user. Amongst the frequently observed syntactic error types in non-native English which it has been argued are attributable to language transfer are subject-verb disagreement, noun-number disagreement, and misuse of determiners. Contrastive analysis was largely displaced in SLA by *error analysis* (Corder, 1967), which argued that there are many other types of error in SLA, and that too much emphasis was placed on transfer errors. However, looking at the relationship in the reverse direction and in a probabilistic manner, contrastive analysis could still be useful to predict the native language of an author through errors found in the text.

The structure of this paper is twofold. Firstly, we explore the potential of some syntactic errors derived from contrastive analysis as useful features in determining the authors' native language: in particular, the three common types of error mentioned above. In other words, we are exploring the contrastive analysis hypothesis in a reverse direction.

Secondly, our study intends to investigate whether such syntactic features are useful stylistic markers for native language identification in addition to other features from the work of Koppel et al. (2005).

The rest of the paper is structured as follows. Section 2 reviews the literature studying native language identification and contrastive analysis. Section 3 decribes the methodology adopted in our study. The experimental results obtained are organised into two separate sections: Section 4 presents the results obtained merely from syntactic features; Section 5 discusses a replication of the work of Koppel et al. (2005) and details the results from adding in the syntactic features. Finally, Section 6 concludes.

2 Literature Review

2.1 Native language identification

Koppel et al. (2005) took a machine learning approach to the task, using as features function words, character n-grams, and part-of-speech (POS) bigrams; they gained a reasonably high classification accuracy of 80% across five different groups of non-native English authors (Bulgarian, Czech, French, Russian, and Spanish), selected from the first version of *International Corpus of Learner English* (ICLE). Koppel et al. (2005) also suggest that syntactic errors might be useful features, but these were not explored in their study. Tsur and Rappoport (2007) replicate this work of Koppel et al. (2005) and hypothesise that the choice of words in second language writing is highly influenced by the frequency of native language syllables – the *phonology* of the native language. Approximating this by character bigrams alone, they achieved a classification accuracy of 66%.

Native language is also among one of the characteristics investigated in the authorship profiling task of Estival et al. (2007). Unlike the approach of Koppel et al. (2005), linguistic errors in written texts are not of concern here; rather this study focuses merely on lexical and structural features. The approach deployed yields a relatively good classification accuracy of 84% when the native language alone is used as the profiling criterion. However, it should be noted that a smaller number of native language groups were examined in this study – namely, Arabic, English, and Spanish. The work was also carried out on data that is not publicly available.

Another relevant piece of research is that of van Halteren (van Halteren, 2008), which has demonstrated the possibility of identifying the source language of medium-length translated texts (between 400 and 2500 words). On the basis of frequency counts of word-based n-grams, surprisingly high classification accuracies from 87% to 97% are achievable in identifying the source language of *European Parliament* (EUROPARL) speeches. Six common European languages were examined – English, German, French, Dutch, Spanish, and Italian. In addition, van Halteren also uncovered salient markers for a particular source language. Many of these were tied to the content and the domain (e.g. the greeting to the European Parliament is always translated a particular way from German to English in comparison with other languages), suggesting a reason for the high classification accuracy rates.

2.2 Contrastive analysis

The goal of contrastive analysis is to predict linguistic difficulties experienced during the acquisition of a second language; as formulated by Lado (1957), it suggests that difficulties in acquiring a new (second) language are derived from the differences between the new language and the native (first) language of a language learner. In this regard, errors potentially made by learners of a second language are predicted from interference by the native language. Such a phenomenon is usually known as *negative transfer*. In error analysis (Corder, 1967), this was seen as only one kind of error, *interlanguage* or *interference errors*; other types were *intralingual* and *developmental* errors, which are not specific to the native language (Richards, 1971).

To return to contrastive analysis, numerous studies of different language pairs have already been carried out, in particular focusing on learners of English. Dušková (1969) investigated Czech learners of English in terms of various lexical and syntactical errors; Light and Warshawsky (1974) examined Russian learners of English (and French learners to some extent) on their improper usage of syntax as well as semantics; Guilford (1998) specifically explored the difficulties of French learners of English in various aspects, from lexical and syntactical to idiosyncratic; and Mohamed et al. (2004) targeted

grammatical errors of Chinese learners in English. Among these studies, commonly observed syntactic error types made by non-native English learners include subject-verb disagreement, noun-number disagreement, and misuse of determiners.

There are many other studies examining interlanguage errors, generally restricted in their scope of investigation to a specific grammatical aspect of English in which the native language of the learners might have an influence. To give some examples, Granger and Tyson (1996) examined the usage of connectors in English by a number of different native speakers – French, German, Dutch, and Chinese; Vassileva (1998) investigated the employment of first person singular and plural by another different set of native speakers – German, French, Russian, and Bulgarian; Slabakova (2000) explored the acquisition of telicity marking in English by Spanish and Bulgarian learners; Yang and Huang (2004) studied the impact of the absence of grammatical tense in Chinese on the acquisition of English tense-aspect system (i.e. telicity marking); Franck et al. (2002) and Vigliocco et al. (1996) specifically examined the usage of subject-verb agreement in English by French and Spanish, respectively.

3 Methodology

3.1 Data

The data used in our study is adopted from the *International Corpus of Learner English* (ICLE) compiled by Granger et al. (2009) for the precise purpose of studying the English writings of non-native English learners from diverse countries. All the contributors to the corpus are believed to possess similar English proficiency level (ranging from intermediate to advanced English learners) and are of about the same age (all in their twenties). This was also the data used by Koppel et al. (2005) and Tsur and Rappoport (2007), although where they used the first version of the corpus, we use the second.

The first version contains 11 sub-corpora of English essays contributed by students of different native languages – Bulgarian, Czech, Dutch, Finnish, French, German, Italian, Polish, Russian, Spanish, and Swedish; the second has been extended to additional 5 other native languages – Chinese, Japanese, Norwegian, Turkish, and Tswana. In this work, we use the five languages of Koppel et al. (2005) – Bulgarian, Czech, French, Russian, Spanish – as well as Chinese and Japanese, based on the work discussed in Section 2.2. For each native language, we randomly select from among essays with length of 500-1000 words: 70 essays for training, 25 essays for testing, and another 15 essays for development. By contrast, Koppel et al. (2005) took all 258 texts from their version for each language and evaluated by ten-fold cross validation. We used fewer with a view to reserving more for future work. From our sample, the average text length broken down by native language is given in Table 1.

Bulgarian	668
Czech	747
French	639
Russian	692
Spanish	621
Chinese	570
Japanese	610

Table 1: Mean text length of native language (words)

3.2 Tools

As in the work discussed in Section 2.1, we use a machine learner. Since its performance in classification problems and its ability in handling high dimensional feature spaces have been well attested (Joachims, 1998), the support vector machine (SVM) is chosen as the classifier. We adopt the online SVM tool, *LIBSVM*[1] (Version 2.89) by Chang and Lin (2001). All the classifications are first conducted under the default settings, where the radial basic function (RBF) kernel is used as it is appropriate for learning a non-linear relationship between multiple features. The kernel is tuned to find the best pair of parameters (C, γ) for data training.

In addition to the machine learning tool, we require a grammar checker that help in detecting the syntactic errors. *Queequeg*,[2] a very small English grammar checker, detects the three error types that are of concern in our study, namely subject-verb disagreement, noun-number disagreement, and misuse of determiners (mostly articles).

4 Syntactic Features

Given that the main focus of this paper is to uncover whether syntactic features are useful in determining

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm

[2] http://queequeg.sourceforge.net/index-e.html

the native language of the authors, syntactic features are first examined separately. Statistical analysis is performed to gain an overview of the distribution of the syntactic errors detected from seven groups of non-native English users. A classification with SVM is then conducted to investigate the degree to which syntactic errors are able to classify the authors according to their native language.

4.1 Features

For the present study, only the three major syntactic error types named above are explored and are used as the syntactic features for classification learning.

Subject-verb disagreement: refers to a situation in which the subject of a sentence disagrees with the verb of the sentence in terms of number or person. An excerpt adopted from the training data that demonstrates such an error: **If the situation become worse . . . /If the situation becomes worse . . . .*

Noun-number disagreement: refers to a situation in which a noun is in disagreement with its determiner in terms of number. An excerpt adopted from the training data that demonstrates such an error: **They provide many negative image . . . /They provide many negative images . . . .*

Misuse of determiners: refers to situations in which the determiners (such as articles, demonstratives, as well as possessive pronouns) are improperly used with the nouns they modify. These situations include missing a determiner when required as well as having an extra determiner when not needed. An excerpt adopted from the training data that demonstrates such an error: **Cyber cafes should not be located outside airport. /Cyber cafes should not be located outside an airport.*[3]

Table 2 provides an overview of which of these grammatical phenomena are present in each native language. All three exist in English; a '-' indicates that generally speaking it does not exists or exists to a much lesser extent in a particular native language (e.g. with Slavic languages and determiners). A '+' indicates that the phenomenon exists, but not that it coincides precisely with the English one. For example, Spanish and French have much more extensive use of determiners than in English; the presence or absence of determiners in Bulgarian has no effect on aspectual interpretation, unlike in English; and as for Chinese and Japanese, the usage of determiners is far less frequent than that of the other languages and generally more deictic in nature. Conjugations (and consequently subject-verb agreement), on the other hand, are more extensive in the European languages than in English.

[3]Such an error may also be recognised as noun-number disagreement in which the grammatical form is . . . *outside airports*; but *Queequeg* identifies this as misuse of determiners.

Language	Subject-verb agreement	Noun-number agreement	Use of determiners
Bulgarian	+	+	+
Czech	+	+	-
French	+	+	+
Russian	+	+	-
Spanish	+	+	+
Chinese	-	-	+
Japanese	-	-	+

Table 2: Presence or absence of grammatical features

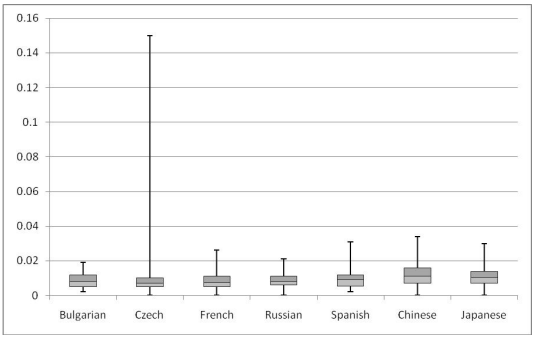

Figure 1: Boxplot: subject-verb disagreement errors

4.2 Data analysis

Boxplots: Figures 1 to 3 depict the distribution of each error type as observed in the training data – 490 essays written by 7 distinct groups of non-native English users. The frequencies of each error type presented in these figures are normalised by the corresponding text length (i.e. the total number of

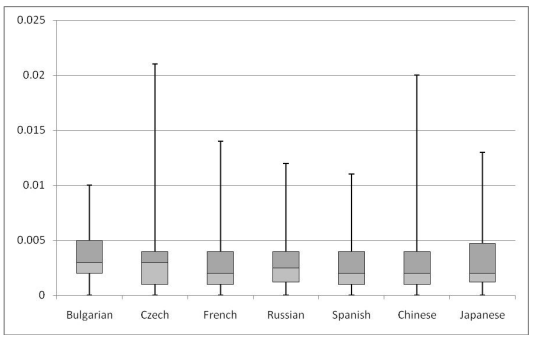

Figure 2: Boxplot: noun-number disagreement errors

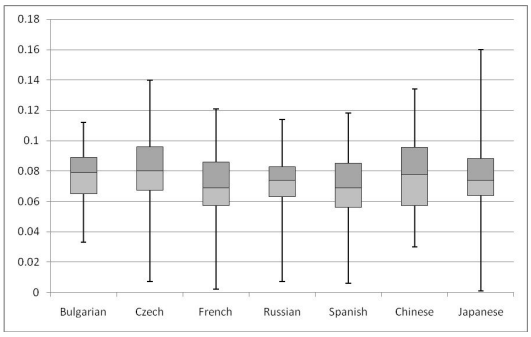

Figure 3: Boxplot: determiner misuse errors

Frequency type	Subject-verb disagreement	Noun-number disagreement	Misuse of determiners
Absolute	0.038	0.114	5.306E-10
Relative	0.178	0.906	0.006

Table 3: P-value of ANOVA test per error type

words). The boxplots present the median, quartiles and range, to give an initial idea of the distribution of each error type.

These boxplots do show some variability among non-native English users with different native languages with respect to their syntactic errors. This is most obvious in Figure 3, with the distribution of errors concerning misuse of determiners. This could possibly be explained by the interference of native language as indicated in the contrastive analysis. Czech and Chinese seem to have more difficulties when dealing with determiners as compared to French and Spanish, since determiners (especially articles) are absent from the language system of Czech and are less frequently used in Chinese, while the usage of determiners in French and Spanish is somewhat different from (and generally more extensive than) in English.

ANOVA tests: The boxplots do not suggest an extremely non-Gaussian distribution, so we use ANOVA tests to determine whether the distributions do in fact differ. A single-factor ANOVA, with the language type being the factor, was carried out for each syntactic error type, for both absolute frequency and relative frequency (normalised by text length). The results are presented in Table 3. Tables 4 to 6 present some descriptive statistics for each of the error types in terms of mean, standard deviation, median, first quartile, and third quartile.

The most interesting result is for the case of determiner misuse. This is highly statistically significant

Language	Mean	Std. Dev.	Median	Q1	Q3
Bulgarian	5.829	3.074	6	4	7
	0.0088	0.0042	0.008	0.005	0.012
Czech	5.414	3.268	5	3	7
	0.0106	0.0213	0.007	0.005	0.01
French	5.243	3.272	4	3	6
	0.0083	0.0048	0.0075	0.005	0.011
Russian	6.086	3.247	6	3	8
	0.0088	0.0045	0.008	0.006	0.011
Spanish	5.786	3.438	5	3	8
	0.0093	0.0051	0.009	0.0053	0.012
Chinese	6.757	3.617	6	4	9
	0.0118	0.0063	0.011	0.007	0.016
Japanese	6.857	4.175	6	4	8
	0.0112	0.0063	0.0105	0.007	0.014

Table 4: Descriptive statistics of subject-verb disagreement errors (first row – absolute frequency; second row – relative frequency)

Language	Mean	Std. Dev.	Median	Q1	Q3
Bulgarian	2.086	1.576	2	1	3
	0.0033	0.0025	0.003	0.002	0.005
Czech	2.457	2.250	2	1	4
	0.0033	0.0033	0.003	0.001	0.004
French	1.814	1.6	1	1	3
	0.003	0.0028	0.002	0.001	0.004
Russian	2.157	1.968	2	1	3
	0.003	0.0024	0.0025	0.0013	0.004
Spanish	1.7	1.376	1.5	1	2
	0.0027	0.0023	0.002	0.001	0.004
Chinese	1.671	1.791	1	1	2
	0.003	0.0032	0.002	0.001	0.004
Japanese	1.971	1.810	1	1	3
	0.0033	0.0029	0.002	0.0013	0.0048

Table 5: As Table 4, for noun-number disagreement

for both absolute and relative frequencies (with the p-values of 5.306E-10 and 0.006 respectively). This seems to be in line with our expectation and the explanation above.

As for subject-verb disagreement, significant differences are only observed in absolute frequency (with a p-value of 0.038). The inconsistency in results could be attributed to the differences in text length. We therefore additionally carried out another single-factor ANOVA test on the text length from our sample (mean values are given in Table 1), which shows that the text lengths are indeed different. The lack of a positive result is a little surprising, as Chinese and Japanese do not have subject-verb

Language	Mean	Std. Dev.	Median	Q1	Q3
Bulgarian	51.471	16.258	47.5	40.25	63.75
	0.0771	0.0169	0.079	0.065	0.089
Czech	61.529	23.766	59.5	44	73
	0.082	0.0253	0.08	0.0673	0.096
French	44.286	14.056	45	34	52
	0.0689	0.0216	0.069	0.0573	0.086
Russian	49.343	15.480	48.5	40.25	59
	0.072	0.0182	0.074	0.063	0.083
Spanish	43.9	15.402	43	31.75	53.75
	0.0706	0.0214	0.069	0.056	0.085
Chinese	44.686	15.373	45	33	54.75
	0.0782	0.0252	0.078	0.0573	0.0958
Japanese	46.243	16.616	43.5	36.25	55.75
	0.0768	0.0271	0.074	0.064	0.0883

Table 6: As Table 4, for determiner misuse

agreement, while the other languages do. However, we note that the absolute numbers here are quite low, unlike for the case of determiner misuse.

Noun-number disagreement, however, does not demonstrate significant differences amongst the seven groups of non-native English users (neither for the absolute frequency nor for the relative frequency), even though again the native languages differ in whether this phenomenon exists. Again, the absolute numbers are small.

Perhaps noun-number disagreement is just not an interference error. Instead, it may be regarded as a developmental error according to the notion of error analysis (Corder, 1967). Developmental errors are largely due to the complexity of the (second) language's grammatical system itself. They will gradually diminish as learners become more competent.

We also note at this point some limitations of the grammar checker *Queequeg* itself. In particular, the grammar checker suffers from false positives, in many cases because it fails to distinguish between count nouns and mass nouns. As such, the checker tends to generate more false positives when determining if the determiners are in disagreement with the nouns they modify. An example of such false positive generated by the checker is as follows: *It could help us to save some money* . . . , where *some money* is detected as ungrammatical. A manual evaluation of a sample of the training data reveals a relatively high false positive rate of 48.2% in determiner misuse errors. (The grammar checker also records a false negative rate of 11.1%.) However, there is no evidence to suggest any bias in the errors with respect to native language, so it just seems to act as random noise.

4.3 Learning from syntactic errors

Using the machine learner noted in Section 3.2, the result of classification based on merely syntactic features is shown in Table 7 below. The majority class baseline is 14.29%, given that there are 7 native languages with an equal quantity of test data. Since only three syntactic error types being examined, it is not unreasonable to expect that the accuracy would not improve to too great an extent. Nevertheless, the classification accuracies are somewhat higher than the baseline, approximately 5% (prior tuning) and 10% (after tuning) better when the relative frequency of the features is being examined. The improvement in classification accuracy after tuning is significant at the 95% confidence level, based on a z-test of two proportions.

Baseline	Presence/ absence	Relative frequency (before tuning)	Relative frequency (after tuning)
14.29% (25/175)	15.43% (27/175)	19.43% (34/175)	24.57% (43/175)

Table 7: Classification accuracy for error features

5 Learning from All Features

The second focus of our study is to investigate the effects of combining syntactic features with lexical features in determining the native language of the authors. To do this, we broadly replicate the work of Koppel et al. (2005) which used a machine learning approach with features commonly used in authorship analysis – function words, character n-grams, and POS n-grams. Koppel et al. (2005) also used spelling errors as features, although we do not do that here. Spelling errors would undoubtedly improve the overall classification performance to some extent but due to time constraints, we keep it for future work.

5.1 Features

Function words: Koppel et al. (2005) did not specify which set of function words was used, although they noted that there were 400 words in the set. Consequently, we explored three sets of function words. Firstly, a short list of 70 function words was examined; these function words were used by Mosteller and Wallace (1964) in their seminal work where they successfully attributed the twelve disputed Federalist papers. Secondly, a long list of 363 function words was adopted from Miller et al. (1958) from where the 70 function words used by Mosteller and Wallace (1964) were originally extracted. Considering that Koppel et al.(2005) made use of 400 function words, we then searched for some stop words commonly used in information retrieval to make up a list of close to 400 words – where our third list consists of 398 function words with stop words[4].

Character n-grams: As Koppel et al. (2005) did not indicate which sort of character n-grams

[4]Stop words were retrieved from Onix Text Retrieval Toolkit. `http://www.lextek.com/manuals/onix/stopwords1.html`

was used, we examined three different types: unigram, bi-gram, and tri-gram. The 200 most frequently used character bi-grams and tri-grams were extracted from our training data. As for unigrams, only the 100 most frequently used ones were extracted since there were fewer than 200 unique unigrams. Space and punctuation were considered as tokens when forming n-grams.

POS n-grams: In terms of POS n-grams, Koppel et al. (2005) tested on 250 rare bi-grams extracted from the Brown corpus. In our study, in addition to 250 rare bi-grams from the Brown corpus, we also examined the 200 most frequently used POS bi-grams and tri-grams extracted from our training data. We used the Brill tagger provided by NLTK for our POS tagging (Bird et al., 2009). Having trained on the Brown corpus, the Brill tagger performs at approximately 93% accuracy.

For each of the lexical features, four sets of classification were performed. The data was examined without normalising, with normalising to lowercase, according to their presence, as well as their relative frequency (per text length). (Note that since both the classification results with and without normalising to lowercase are similar, only the results without normalising will be presented.)

5.2 Results

Individual features: The classification results (before tuning) for each lexical feature – function words, character n-grams, and POS n-grams – are presented in Table 8, 9, and 10, respectively. Each table contains results with and without integrating with syntactic features (i.e. the three syntactic error types as identified in Section 4). It is obvious that function words and POS n-grams perform with higher accuracies when their presence is used as the feature value for classification; whereas character n-grams perform better when their relative frequency is considered. Also note that the best performance of character n-grams (i.e. bi-grams) before tuning is far below 60%, as compared with the other two lexical features. It, however, achieves as high as 69.14% after tuning where both function words and POS bi-grams are at 64.57% and 66.29%, respectively.

The classification results for the 250 rare bi-grams from the Brown corpus are not presented here since the results are all at around the baseline (14.29%).

Function words	Presence/ absence (- errors)	Presence/ absence (+ errors)	Relative frequency (- errors)	Relative frequency (+ errors)
70 words	50.86% (89/175)	50.86% (89/175)	40.57% (71/175)	42.86% (75/175)
363 words	60.57% (106/175)	61.14% (107/175)	41.71% (73/175)	43.43% (76/175)
398 words	65.14% (114/175)	65.14% (114/175)	41.71% (73/175)	43.43% (76/175)

Table 8: Classification accuracy for function words

Character n-grams	Presence/ absence (- errors)	Presence/ absence (+ errors)	Relative frequency (- errors)	Relative frequency (+ errors)
Character unigram	56.57% (99/175)	56.57% (99/175)	50.29% (88/175)	42.29% (74/175)
Character bi-gram	22.86% (40/175)	22.86% (40/175)	50.29% (88/175)	41.71% (73/175)
Character tri-gram	28.57% (50/175)	28.57% (50/175)	43.43% (76/175)	30.29% (53/175)

Table 9: Classification accuracy for character n-grams

Combined features: Table 11 presents both before and after tuning classification results of all combinations of lexical features (with and without syntactic errors). Each lexical feature was chosen for combination based on their best individual result. The combination of all three lexical features results in better classification accuracy than combinations of two features, noting however that character n-grams make no difference. In summary, our best accuracy thus far is at 73.71%. As illustrated in the confusion matrix (Table 12), misclassifications occur largely in Spanish and the Slavic languages.

5.3 Discussion

Comparisons with Koppel et al. (2005): Based on the results presented in Table 8 and 9, our classification results prior to tuning for both function words and character n-grams (without considering the syntactic features) appear to be lower than the results obtained by Koppel et al. (2005) (as presented in Table 13). However, character n-grams performs on par with Koppel et al. after tuning. The difference in classification accuracy (function words in particular) can be explained by the corpus size. In our study, we only adopted 110 essays for each native language. Koppel et al. made use of 258 essays for each native language. A simple analysis (extrapo-

POS n-grams	Presence/ absence (- errors)	Presence/ absence (+ errors)	Relative frequency (- errors)	Relative frequency (+ errors)
POS bi-gram	62.86% (110/175)	63.43% (111/175)	58.29% (102/175)	48.0% (84/175)
POS tri-gram	57.71% (101/175)	57.14% (100/175)	48.0% (84/175)	37.14% (65/175)

Table 10: Classification accuracy for POS n-grams

Combinations of features	prior tuning (- errors)	prior tuning (+ errors)	after tuning (- errors)	after tuning (+ errors)
Function words + character n-grams	58.29% (102/175)	58.29% (102/175)	64.57% (113/175)	64.57% (113/175)
Function words + POS n-grams	73.71% (129/175)	73.71% (129/175)	73.71% (129/175)	73.71% (129/175)
Character n-grams + POS n-grams	63.43% (111/175)	63.43% (111/175)	66.29% (116/175)	66.29% (116/175)
Function words + char n-grams + POS n-grams	72.57% (127/175)	72.57% (127/175)	73.71% (129/175)	73.71% (129/175)

Table 11: Classification accuracy for all combinations of lexical features

	BL	CZ	FR	RU	SP	CN	JP
BL	[16]	4	-	5	-	-	-
CZ	3	[18]	-	3	1	-	-
FR	1	-	[24]	-	-	-	-
RU	3	4	3	[14]	-	-	1
SP	1	2	4	3	[14]	-	1
CN	1	1	1	-	-	[20]	2
JP	-	-	-	-	-	4	[21]

Table 12: Confusion matrix based on both lexical and syntactic features (BL:Bulgarian, CZ:Czech, FR:French, RU:Russian, SP:Spanish, CN:Chinese, JP:Japanese)

lating from a curve fitted by a linear regression of the results for variously sized subsets of our data) suggests that our results are consistent with Koppel et al.'s given the sample size. (Note that the results of POS n-grams could not be commented here since Koppel et al. had considered these features as errors and did not provide a separate classification result.)

Usefulness of syntactic features: For the best combinations of features, our classification results of integrating the syntactic features (i.e. syntactic error types) with the lexical features do not demonstrate any improvement in terms of classification accuracy. For the individual feature types with results in Table 8 to Table 10, the syntactic error types sometimes in fact decrease accuracies. This could be due to the small number of syntactic error types being considered at this stage. Such a small number of features (three in our case) would not be sufficient to add much to the approximately 760 features used in our replication of the Koppel et al.'s work. Furthermore, error detection may be flawed as the result of the limitations noted in the grammar checker.

Other issues of note: Character n-grams, as seen in our classification results (see Table 11) do not seem to be contributing to the overall classification.

Types of lexical feature	Koppel et al.	Our best result (prior tuning)	Our best result (after tuning)
Function words	~71.0%	~65.0%	~65.0%
Character n-grams	~68.0%	~56.0%	~69.0%

Table 13: Comparison of results with Koppel et al.

It is noticeable when character n-grams are combined with function words and when combined with POS n-grams separately. Both combinations do not exhibit any improvement in accuracy. In addition, with character n-grams adding to the other two lexical features, the overall classification accuracy does not seem to be improved either. Nevertheless, as mentioned in Section 5.2 (under individual features), character n-grams alone are able to achieve an accuracy close to 69%. It seems that character n-grams are somehow a useful marker as argued by Koppel et al. (2005) that such feature may reflect the orthographic conventions of individual native language. Furthermore, this is consistent with the hypothesis put forward by Tsur and Rappoport (2007) in their study. It was claimed that the choice of words in second language writing is highly influenced by the frequency of native language syllabus (i.e. the *phonology* of the native language) which can be captured by character n-grams. For example, confusion between phonemes /l/ and /r/ is commonly observed in Japanese learners of English.

6 Conclusion

We have found some modest support for the contention that contrastive analysis can help in detecting the native language of a text's author, through a statistical analysis of three syntactic error types and through machine learning using only features based on those error types. However, in combining these with features used in other machine learning approaches to this task, we did not find an improvement in classification accuracy.

An examination of the results suggests that using more error types, and a method for more accurately identifying them, might result in improvements. A still more useful approach might be to use an automatic means to detect different types of syntactic errors, such as the idea suggested by Gamon (2004) in which context-free grammar production rules can be explored to detect ungrammatical structures based on long-distance dependencies. Furthermore, error analysis may be worth exploring to uncover non-interference errors which could then be discarded as irrelevant to determining native language.

Acknowledgments

The authors would like to acknowledge the support of ARC Linkage grant LP0776267, and thank the reviewers for useful feedback.

References

Stephen Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Stephen P. Corder. 1967. The significance of learners' errors. *International Review of Applied Linguistics in Language Teaching (IRAL)*, 5(4):161–170.

Libuše Dušková. 1969. On sources of error in foreign language learning. *International Review of Applied Linguistics (IRAL)*, 7(1):11–36.

Dominique Estival, Tanja Gaustad, Son-Bao Pham, Will Radford, and Ben Hutchinson. 2007. Author profiling for English emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 263–272.

Ian Fette, Norman Sadeh, and Anthony Tomasic. 2007. Learning to detect phishing emails. In *Proceedings of the 16th International World Wide Web Conference*.

Julie Franck, Gabriella Vigliocco, and Janet Nicol. 2002. Subject-verb agreement errors in French and English: The role of syntactic hierarchy. *Language and Cognitive Processes*, 17(4):371–404.

Michael Gamon. 2004. Linguistic correlates of style: Authorship classification with deep linguistic analysis features. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 611–617.

Sylviane Granger and Stephanie Tyson. 1996. Connector usage in the English essay writing of native and non-native EFL speakers of English. *World Englishes*, 15(1):17–27.

Sylviane Granger, Estelle Dagneaux, Fanny Meunier, , and Magali Paquot. 2009. *International Corpus of Learner English (Version 2)*. Presses Universitaires de Louvain, Louvain-la-Neuve.

Jonathon Guilford. 1998. English learner interlanguage: What's wrong with it? *Anglophonia French Journal of English Studies*, 4:73–100.

Thorsten Joachims. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In *Machine Learning: ECML-98*, pages 137–142.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Automatically determining an anonymous author's native language. In *Intelligence and Security Informatics*, volume 3495 of *Lecture Notes in Computer Science*, pages 209–217. Springer-Verlag.

Robert Lado. 1957. *Linguistics Across Cultures: Applied Linguistics for Language Teachers*. University of Michigan Press, Ann Arbor, MI, US.

Richard L. Light and Diane Warshawsky. 1974. Preliminary error analysis: Russians using English. Technical report, National Institute of Education, USA.

George A. Miller, E. B. Newman, and Elizabeth A. Friedman. 1958. Length frequency statistics for written English. *Information and Control*, 1(4):370–389.

Abdul R. Mohamed, Li-Lian Goh, and Eliza Wan-Rose. 2004. English errors and Chinese learners. *Sunway College Journal*, 1:83–97.

Frederick Mosteller and David L. Wallace. 1964. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley, Reading, MA, US.

Jack C. Richards. 1971. A non-contrastive approach to error analysis. *ELT Journal*, 25(3):204–219.

Roumyana Slabakova. 2000. L1 transfer revisited: the L2 acquisition of telicity marking in English by Spanish and Bulgarian native speakers. *Linguistics*, 38(4):739–770.

Oren Tsur and Ari Rappoport. 2007. Using classifier features for studying the effect of native language on the choice of written second language words. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 9–16.

Hans van Halteren. 2008. Source language markers in EUROPARL translations. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, pages 937–944.

Irena Vassileva. 1998. Who am I/how are we in academic writing? A contrastive analysis of authorial presence in English, German, French, Russian and Bulgarian. *International Journal of Applied Linguistics*, 8(2):163–185.

Garbriella Vigliocco, Brian Butterworth, and Merrill F. Garrett. 1996. Subject-verb agreement in Spanish and English: Differences in the role of conceptual constraints. *Cognition*, 61(3):261–298.

Suying Yang and Yue-Yuan Huang. 2004. The impact of the absence of grammatical tense in L1 on the acquisition of the tense-aspect system in L2. *International Review of Applied Linguistics in Language Teaching (IRAL)*, 42(1):49–70.

Rong Zheng, Yi Qin, Zan Huang, and Hsinchun Chen. 2003. Authorship analysis in cybercrime investigation. In *Intelligence and Security Informatics*, volume 2665 of *Lecture Notes in Computer Science*, pages 59–73. Springer-Verlag.

Faster parsing and supertagging model estimation

Jonathan K. Kummerfeld[a] **James R. Curran**[a] **Jessika Roesner**[b]
School of Information Technologies[a] Department of Computer Science[b]
University of Sydney University of Texas at Austin
NSW 2006, Australia Austin, TX, USA
{jkum0593,james}@it.usyd.edu.au jessi@mail.utexas.edu

Abstract

Parsers are often the bottleneck for data acquisition, processing text too slowly to be widely applied. One way to improve the efficiency of parsers is to construct more confident statistical models. More training data would enable the use of more sophisticated features and also provide more evidence for current features, but gold standard annotated data is limited and expensive to produce.

We demonstrate faster methods for training a supertagger using hundreds of millions of automatically annotated words, constructing statistical models that further constrain the number of derivations the parser must consider. By introducing new features and using an automatically annotated corpus we are able to double parsing speed on Wikipedia and the Wall Street Journal, and gain accuracy slightly when parsing Section 00 of the Wall Street Journal.

1 Introduction

Many systems in NLP for tasks such as Question–Answering rely on large volumes of data. Parsers are a useful means of extracting extra information about text, providing the syntactic structure of sentences. However, when they are the bottleneck in the data acquisition phase of a system simple solutions are to use less data, or not use a parser at all. If we can improve the speed of parsers this will be unnecessary.

For lexicalised grammars such as Combinatory Categorial Grammar (CCG) (Steedman, 2000) the step in which words are labelled with lexical categories has great influence on parsing speed and accuracy. In these formalisms, the labels chosen constrain the set of possible derivations so much that the process of choosing them, *supertagging*, is described as 'almost parsing' (Joshi and Bangalore, 1994). If the supertagger is more accurate it can further constrain the set of possible derivations by supplying fewer categories, leaving the parser with less to do.

One means of improving the supertagger's statistical model of language is to provide more evidence, in this case, more annotated text. However, creating a significant amount of extra gold standard annotated text is not feasible. An alternative approach is 'semi-supervised training', in which a small set of annotated data and a much larger set of unannotated data is used. Training a system directly on its own output, 'self-training', is not normally effective (Clark et al., 2003), but recently McClosky et al. (2006) demonstrated that parser output can be made useful for retraining by the application of a reranker.

To enable the use of more training data we have parallelised the C&C parser's supertagger training process and implemented perceptron–based algorithms for parameter estimation. In the process of this work we also modified the C&C parser's use of a particular CCG rule, based on observations of its behaviour. Our unlabeled training data was part of the English section of Wikipedia, consisting of 47 million sentences. We used the C&C parser to label the data with supertags, producing training data that could then be used to retrain its supertagger. The reasoning behind the use of this data is that the supertagger will provide categories that the parser is most likely to use in a spanning analysis.

Models trained on WSJ and Wikipedia data parsed sentences up to twice as fast, without decreasing accuracy. And the perceptron–based algorithms enabled the use of much larger data sets, without loss in parsing speed or accuracy.

2 Background

Parsing is the process of analysing a set of tokens and extracting syntactic structure. In the context of natural language we are faced with several challenges, including ambiguous sentences that are context sensitive and a grammar that is unknown and constantly changing.

Two main classes of grammars have been used to try to understand and model natural language, phrasal and lexicalised grammars. Phrasal grammars generally define a small set of labels that capture the syntactic behaviour of a word in a sentence, such as noun and adverb, and then use a large set of rules to construct a phrase structure tree in which the leaves are the words and the internal nodes are applications of rules. Lexicalised grammars provide a much larger set of categories for lexical items and only a few rules. The categories provide a more detailed description of a word's purpose in a sentence, while the rules are simple descriptions of how pairs of categories can combine to form the parse tree.

We use the lexicalised grammar formalism Combinatory Categorial Grammar (CCG) (Steedman, 2000). In CCG, there are two types of categories, *atomic*, which are one of S, N, NP and PP, and *complex*, which contain two parts, an argument and a result, denoted by either 'Result / Argument' or 'Result \ Argument', where the slashes indicate whether the Argument is expected to lie to the right or left respectively, and the result and argument are categories themselves. To form a derivation for English sentences these categories are combined according to seven rules, forward and backward application, forward and backward composition, backward crossed composition, type raising and coordination.

Figure 1 presents two example CCG derivations. In both examples, the line directly beneath the words contains the category that was assigned to each word, NP for 'I', (S\NP)/NP for 'ate' and so on. The lines that follow show a series of rule applications, building up the parse tree.

The lines with a $>$ sign at the end indicate forward application, which occurs when a complex category is of the form 'Result / Argument' and its argument is the same as the category to its right. The lines with a $<$ sign at the end are instances of backward application, which works in the same way, but in the opposite direction.

Note in particular the change of tag for 'with' in the two examples and its affect on the subsequent rule applications. The decision made by the supertagger effectively decides which analysis will be found, or if both are provided the parser must consider more possible derivations.

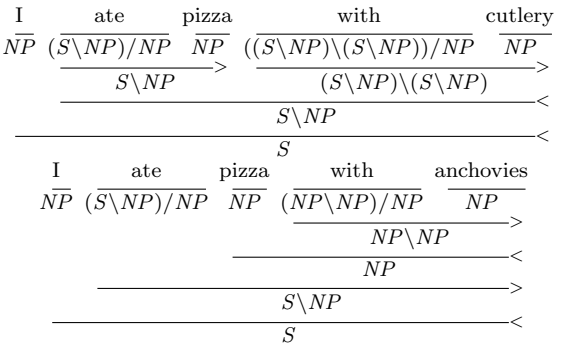

Figure 1: Two CCG derivations with PP ambiguity.

The CCG parser and associated supertagger we have used is the C&C parser (Clark and Curran, 2003; Clark and Curran, 2007b). The supertagger applies categories to words using the forward backward algorithm, and the parser forms a derivation by applying the Cocke–Younger–Kasami (CKY) chart parsing algorithm (Younger, 1967; Kasami, 1967) and dynamic programming.

2.1 Supertagging

Supertags were first proposed by Joshi and Bangalore (1994) for Lexicalized Tree-Adjoining Grammar (LTAG). Like POS tags, supertags are assigned to each word in the sentence prior to parsing, but supertags contain much more detailed syntactic information. This leads to tag sets that are up to two orders of magnitude larger. The first supertaggers gave each word a single tag based only on the POS tags in the local context and had an accuracy below 90% (Chandrasekar and Bangalore, 1997b). While this is not accurate enough for incorporation into a wide–coverage parser, it was enough to be useful in an information retrieval system (Chandrasekar and Bangalore, 1997a), attaining an F-score of 92% for filtering out irrelevant documents. Accuracy was improved by the use of multitaggers (Chen et al., 1999), but as more tags are supplied the parsing efficiency decreases (Chen et al., 2002), demon-

strating that lexical ambiguity is an important factor in parsing complexity (Sarkar et al., 2000).

Supertagging was first applied to CCG by Clark (2002). Rather than defining a fixed number of tags to be produced per word, the CCG supertagger includes all tags with probabilities within some factor, β, of the most probable tag. Also, during parsing the β value starts high, and if a derivation is not found it is progressively decreased. This provides similar speed benefits to a single tagger, but without a loss in coverage. Previous attempts to expand the feature set used by the CCG supertagger were unsuccessful because of data sparseness issues (Cooper, 2007).

Perhaps the closest previous work was by (Sarkar, 2007), who incorporated a supertagger with a full LTAG parser, and demonstrated improved efficiency through the use of training data annotated by the parser. This led to higher performance than entirely supervised training methods.

2.2 Semi-supervised training

One of the first demonstrations of semi-supervised training in NLP was the use of a 'co-training' method by Yarowsky (1995), who achieved 96% accuracy on a word sense disambiguation task. A similar method was subsequently applied to statistical parsing by Sarkar (2001), leading to a 9% increase in F-score for an LTAG parser.

Co-training relies upon two independent views of the data to construct models that can inform each other. Another method of semi-supervised training is to apply a re-ranker to the output of a system to generate new training data. By applying a re-ranker to the output of a parser McClosky et al. (2006) were able to improve on the best result for Wall Street Journal parsing by 0.8%, but with no significant change in efficiency.

2.3 Perceptron Algorithms

The perceptron is an online classification method that was proposed by Rosenblatt (1958). However, the algorithm only converges for linearly separable datasets. Recently Freund and Schapire (1999) developed the Averaged Perceptron (AP), which stores all weight vectors during training and combines them in a weighted majority vote to create the final weight vector. This variation led to performance competitive with modern techniques, such as Support Vector Machines, on a handwritten digit classification task.

Another recent variation is the Margin Infused Relaxed Algorithm (MIRA) (Crammer and Singer, 2003), which adjusts the weight vector only enough to cause the current instance to be correctly classified with a specified margin. This method generally has a lower relative error than the standard perceptron but makes more updates.

Collins (2002) showed that applying these methods to tasks in NLP produced better performance than maximum entropy models. Specifically, using a voted perceptron and trigram features for training, a Viterbi based system achieved an F-score of 93.53% for NP Chunking and an error rate of 2.93% for POS tagging, compared to 92.65% and 3.28% respectively for a similar system trained with a maximum entropy model.

Collins and Roark (2004) applied these methods to parsing, using an incremental beam search parser. The parser performed similarly to another based on a generative model, with an F-score of 87.8% for data with gold standard POS tags, and 86.6% for tags generated by a tagger. Similar methods were recently applied to the C&C parser (Clark and Curran, 2007a), leading to similar performance to a log-linear model, but with much lower system requirements. Zhang et al. (2009) used the averaged perceptron algorithm to train an HPSG supertagger, with similar improvements to training time as described here.

3 Implementation

The parser uses the CKY algorithm to construct the 'chart', an efficient representation of all possible analyses for a sentence. The most probable derivation is found using the Viterbi algorithm and probabilities are calculated based on a conditional log-linear model.

The supertagger uses a maximum entropy based model to assign a set of possible lexical categories to each word in the sentence. The main aspect of the tagging process relevant to this work is the role of beta levels.

If the supertagger assigns only one category to each word its accuracy is too low to be effectively incorporated into a parser. By multitagging we can make the supertagger more accurate, but

at the cost of speed as the parser must consider larger sets of possible categories. The beta levels define cutoffs for multitagging based on the probabilities from the maximum entropy model. If the parser is unable to form a spanning analysis the beta level is decreased and the supertagger is rerun to retrieve larger sets of supertags.

These levels have a large influence on parsing accuracy and speed. Accuracy varies because the set of possible derivations increases as more tags are supplied, leading the parser to choose different derivations at different levels. Speed varies as the time spent attempting to form a parse increases as more tags are supplied. Also, for sentences that are not parsed at the first level each attempt at another level requires more time.

The initial feature set used for tagging included unigrams of POS tags and words and bigrams of POS tags, all in a five word window surrounding the word being tagged. The weights for these features were estimated on a single CPU using either Generalised Iterative Scaling (GIS) (Darroch and Ratcliff, 1972) or the Broyden-Fletcher-Goldfarb-Shanno method (BFGS) (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970). Here we consider two other algorithms, a parallelised form of the process, and a range of extra features.

3.1 Averaged Perceptron

The standard multi-class perceptron maintains a matrix of weights, containing a row for each attribute and a column for each class. When all attributes are binary valued the class is assigned by ignoring all rows for attributes that do not occur and determining which column has the greatest sum. During training the class that corresponds to the column with the greatest sum is compared to the true class and if it is correct no change is made. If the predicted class is incorrect the weights are updated by subtracting 1.0 from all weights for the predicted class and adding 1.0 to all weights for the true class. The averaged perceptron follows the same algorithm, but returns the average of the weight matrix over the course of training, rather than its final state.

3.2 Margin Infused Relaxed Algorithm

MIRA also follows the standard multi-class perceptron algorithm, but applies a different update method. The intention is to make the smallest change to the weights such that the correct class is produced by a given margin. We use a slight variation of the update function defined by Crammer and Singer (2003), expressed as follows:

$$\min\left(\max, \frac{\text{margin}+\sum_f p_w - t_w}{|\text{features}|(1+\frac{1}{n_{above}})}\right)$$

where $margin$ is the absolute difference that will be created between the true classification and those that previously ranked above it, the sum is over all features, p_w and t_w are the weights associated with the feature f for the predicted and true classes respectively, $|features|$ is the number of active features, and n_{above} is the number of categories that had higher sums than the correct category. The constant max is introduced to prevent a single event causing extremely large changes to the model.

We have also applied shuffling between iterations of the algorithm to prevents the model from overfitting to the particular order of training instances.

3.3 Parallelisation

To enable the use of more data and features we increased the amount of accessible RAM and processing power by parallelising the supertagger training using the Message Passing Interface (MPI) and the MapReduce library MRMPI[1].

The first stages of supertagging are feature extraction and aggregation. Extraction was parallelised by dividing the data amongst a set of computers and having each one extract the features in its set. Aggregation is necessary to determine overall frequencies for features, and to reorder the features to maximise efficiency. For the aggregation process we used the MRMPI library.

For weight estimation using maximum entropy methods the main calculations are sums of weights across all training instances. The parallel versions of GIS and BFGS differ in three main ways. First, the data is divided between a set of computers. Second, sums are calculated across all computers to determine necessary changes to weights. And third, after each update the changes are distributed to all nodes.

[1] http://www.sandia.gov/ sjplimp/mapreduce.html

The perceptron methods adjust the weights based on each training instance individually and so the parallelisation above was not applicable. The training instances are still distributed across a cluster of computers, but only one computer is working at a time, adjusting the weights based on each of its instances and then passing the weights to the next node. This saves time by removing the cost of loading the training instances from disk when there are too many to fit in RAM.

3.4 Blocking Excess Backward Composition

In the process of debugging the parser, we investigated the number of times particular pairs of categories were combined. We were surprised to discover that a very large number of backward compositions were being performed in the chart, even though backward composition rarely occurred in the parser output (or in the gold standard itself).

Backward composition is normally used for non-constituent coordination between pairs of type-raised categories, but the parser was also using it for combining non-type-raised and type-raised categories. This is an instance where the Eisner (1996) normal form constraints have failed to stop non-normal form derivations, because Eisner's constraints were not designed to work with type-raising. We added a constraint that only allows backward composition to occur if both children are type-raised.

4 Methodology

4.1 Data

Evaluation has been performed using Section 00 of CCGBank, a translation of the Penn Treebank to CCG (Hockenmaier, 2003). Sections 02-21 were used as training data and are simply referred to as WSJ in the following section. The raw Wikipedia data was tokenised using Punkt (Kiss and Strunk, 2006) and the NLTK tokeniser (Bird et al., 2009), and parsed using the C&C parser and models version 1.02[2]. The WSJ sentences had an average length of 23.5 words and a variance of 122.0 while the Wikipedia sentences had an average length of 21.7 words and a variance of 151.0.

[2]http://svn.ask.it.usyd.edu.au/trac/candc

4.2 Evaluation

For a given beta level the number of categories assigned to each word by the supertagger will vary greatly between models. This presents a problem because as described in Section 3 the number of categories assigned has a large influence on parsing speed and accuracy. To fairly compare the models presented here we have tuned all five beta levels on the test set to ensure all models assign the same number of categories per word on average. When testing on Wikipedia text we have used the same beta levels as for WSJ and included the ambiguity this leads to in the tables of results.

F-scores are calculated based on comparisons with gold standard labelled dependencies for Section 00. Category accuracies (Cat.) are for the first beta level only, and are the percentage of words in the sentence that were assigned a tagset that includes the correct category. Category accuracy for Wikipedia was measured over three hundred Wikipedia sentences that were hand–annotated with supertags and grammatical relations, containing 6696 word–category pairs (Clark et al., 2009).

Statistical significance testing was used to determine if changes in performance were meaningful. The test applied reports whether two sets of responses are drawn from the same distribution, where scores of 0.05 and lower are considered significant (Chinchor, 1992).

To measure parsing speed we used ten thousand unseen WSJ sentences from 1988 and ten thousand unseen Wikipedia sentences. The WSJ set was chosen as it is similar to the CCGBank WSJ evaluation set, but much larger and so the per sentence speed should be more accurate. The Wikipedia set is used as the two domains contain different writing styles, meaning the use of Wikipedia based self-training data should lead to particular improvement in speed on that form of text. The datasets only contain sentences of at least six and at most two hundred and fifty tokens.

As the amount of training data scales up, so too does the time it takes to train models. To demonstrate the benefits of perceptron based techniques we measured the amount of time models take to train. These measurements were performed using a 3GHz Intel Core 2 Duo CPU, and 4Gb of RAM.

	Accuracy (%)		Speed	
	WSJ		WSJ	Wiki
Parser	Cat.	F	(sent / sec)	
C&C 1.02	96.07	83.22	31.7	30.8
Modified	96.07	83.41	47.8	45.8

Table 1: The effect of introducing extra constraints on the use of backward composition on speed and accuracy. The supertagging model was constructed using BFGS and sections 02–21 of the WSJ.

5 Results

In this section we present four sets of experiments. First, the change in backward composition handling is evaluated, comparing the speed and accuracy of the standard model before and after the change. Second we consider the benefits of larger data sets, training models using the same algorithm but a range of training sets. Next we compare the new estimation algorithms described above with GIS and BFGS. Finally we explore the impact of introducing extra features.

5.1 Modified Backward Composition

The influence of the change to backward composition handling is shown in Table 1. A clear speed increase of more than 45% is achieved, and a statistically significant increase in F-score occurred.

5.2 Training Data Type and Volume

To investigate the effectiveness of semi-supervised training we constructed a series of models using the GIS algorithm and a selection of datasets. In Table 5.2 we can see that the use of Wikipedia data labelled by the parser causes a clear improvement in parsing speed on Wikipedia. We also observe that when the WSJ accounts for less than 10% of the training set, parsing speed on the WSJ decreases.

It is interesting to compare the baseline model and the models trained on Wikipedia. The model trained on forty thousand Wikipedia sentences only, approximately the same amount of text as in section 02-21 of the WSJ, has much lower supertagging accuracy, but much higher parsing speed. This makes sense as the text the model is trained on is not the true derivation, but rather the derivation that the parser chose. This means that the supertagging model is trained to produce the set of tags that the parser is most likely to combine into its final analysis. As a result, the set assigned at the first beta level is less accurate, but more likely to form a spanning analysis.

	Accuracy (%)			Amb.	Speed	
	WSJ		Wiki	Wiki	WSJ	Wiki
Data	Cat.	F	Cat.		(sent / sec)	
WSJ						
0k	**96.32**	83.82	95.34	1.32	51.7	46.8
Wiki						
40k	93.90	79.83	94.79	1.26	48.1	**61.3**
400k	95.07	81.75	95.71	1.27	46.9	**61.3**
2000k	95.54	82.57	**95.80**	1.28	45.0	57.3
WSJ + Wiki						
40k	96.31	**83.90**	95.37	1.29	**54.3**	58.9
400k	96.22	83.69	95.68	1.28	50.6	59.7
2000k	96.27	83.70	95.73	1.28	47.9	59.7

Table 2: The effect of self-training on supertagging accuracy and parsing F-score. Numbers in the 'Data' column indicate how much Wikipedia text was used.

The decreases in F-score when training on only Wikipedia are statistically significant, while the changes when training on a combination of the WSJ and Wikipedia are not. Interestingly, the models trained on only Wikipedia are also slower when parsing the WSJ than the baseline, and the models trained on a mixture of data are progressively slower as more Wikipedia data is used.

5.3 Algorithm Comparison

Using larger datasets for training can take a prohibitive amount of time for the GIS and BFGS algorithms. However, any time benefits provided by other algorithms need to be balanced with their influence on accuracy. Table 3 shows the results of experiments investigating this trade-off.

It is clear from the training speed column that the perceptron based algorithms, AP and MIRA, train approximately two orders of magnitude faster than GIS and BFGS.

It is also interesting to note the change in the average number of categories assigned for Wikipedia sentences. As expected, the ambiguity level is decreasing as more Wikipedia text is used, but at the same time the tagging accuracy remains fairly constant or improves slightly. This indicates that the automatically labelled data is useful in adapting the supertagger to the Wikipedia domain.

Importantly, the changes in F-score between

	Accuracy (%)			Amb.	Speed		
	WSJ		Wiki	Wiki	Train	WSJ	Wiki
Data	Cat.	F	Cat.		(sec)	(sent / sec)	
WSJ							
GIS	**96.32**	83.82	95.34	1.32	7,200	51.7	46.8
BFGS	96.29	83.73	95.33	1.31	6,300	52.1	48.5
AP	95.65	83.74	94.49	1.35	76	59.2	57.1
MIRA	96.19	83.69	95.19	1.33	96	50.6	47.9
WSJ + 40k Wiki							
GIS	96.31	**83.90**	95.37	1.29	14,000	54.3	58.9
BFGS	96.14	83.86	95.24	1.29	13,000	52.1	60.7
AP	95.68	83.79	94.61	1.28	160	**62.8**	**69.7**
MIRA	96.18	83.77	95.28	1.30	200	54.0	58.6
WSJ + 400k Wiki							
GIS	96.22	83.69	95.68	1.28	*	50.6	59.7
AP	95.77	83.56	95.16	1.27	950	57.8	69.4
MIRA	96.19	83.41	95.58	1.28	1,200	52.3	61.4
WSJ + 2,000k Wiki							
GIS	96.27	83.70	**95.73**	1.28	*	47.9	59.7
MIRA	96.22	83.52	95.62	1.28	*	52.0	59.3

Table 3: Comparison of model estimation algorithms. The models missing times were trained on a different computer with more RAM and are provided for accuracy comparison.

models in each section are not statistically significant. This indicates that the perceptron based algorithms are just as effective as GIS and BFGS.

5.4 Feature Extension

The final set of experiments involved the exploration of extra features. Using the MIRA training method we were able to quickly construct a large set of models, as shown in Table 4.

The standard features used by the supertagger are unigrams of words and unigrams and bigrams of POS tags in a five word window. We considered expansions of this set to include bigrams of words and trigrams of POS tags, and all of the features extended to consider a seven word window, which are indicated by the word 'far'.

The results in the first section of the table, training on the WSJ only, are unsurprising. With such a small amount of data these features are too rare to have a significant impact, and it is likely that they lead the model to over-fit. The best result in this section does not produce a statistically significant improvement over the baseline. However, in the second and third sections of the table the differences between the best models and the baseline are statistically significant. Also, the model with

	Accuracy (%)			Speed	
	WSJ		Wiki	WSJ	Wiki
Features	Cat.	F	Cat.	(sent / sec)	
WSJ					
All	96.25	83.69	95.12	45.1	42.8
- far tags	96.13	83.68	95.15	46.4	42.9
- bitags	96.15	83.84	95.24	45.2	42.1
- far bitags	96.22	83.83	95.24	45.3	43.2
- tritags	96.23	83.79	**95.34**	45.2	42.6
- far tritags	96.22	83.86	95.31	45.5	43.2
- far words	96.28	83.83	95.27	46.2	43.1
- biwords	96.22	83.81	95.19	45.9	45.4
- far biwords	96.26	83.89	95.19	45.5	43.7
- triwords	**96.31**	83.80	95.16	48.0	46.0
- far triwords	96.25	**83.91**	95.24	46.2	43.6
Baseline	96.19	83.69	95.19	**50.6**	**47.9**
WSJ + 40k Wiki					
All	96.29	84.00	95.45	48.7	55.9
- far tags	96.20	83.96	95.45	48.1	53.6
- bitags	96.15	83.84	95.24	45.2	42.3
- far bitags	96.28	**84.17**	95.33	48.3	55.8
- tritags	96.25	83.88	95.47	48.2	54.5
- far tritags	96.34	83.85	**95.49**	49.7	54.9
- far words	96.32	84.04	95.47	48.1	55.7
- biwords	96.31	84.04	95.31	50.5	57.4
- far biwords	96.35	84.10	95.42	49.2	55.0
- triwords	**96.39**	**84.17**	95.42	50.0	57.8
- far triwords	96.32	84.00	95.47	49.6	55.5
Baseline	96.18	83.77	95.28	**54.0**	**58.6**
WSJ + 400k Wiki					
All	96.42	83.80	95.82	48.2	57.4
- far tags	96.38	83.72	95.79	48.3	57.3
- bitags	96.34	83.79	95.85	42.0	56.9
- far bitags	96.34	83.85	95.79	49.4	57.8
- tritags	96.38	83.81	**95.91**	49.2	57.9
- far tritags	96.39	83.94	**95.91**	50.2	56.8
- far words	**96.46**	83.73	**95.91**	48.8	57.4
- biwords	96.35	83.74	95.74	50.0	58.3
- far biwords	96.40	**83.97**	95.82	49.7	57.8
- triwords	96.37	83.96	95.74	49.7	58.8
- far triwords	96.40	83.86	95.83	49.9	58.4
Baseline	96.19	83.41	95.58	**52.3**	**61.4**

Table 4: Subtractive analysis of various feature sets. In each section the category accuracy values that are lower than those for 'All' have been underlined as removing these features decreases accuracy. The bold values are the best in each column for each section. The baseline model uses the default feature set for the C&C parser.

the best result in the table produces a statistically significant improvement in recall over the models in Table 3 constructed using the same data.

6 Future Work

A wide range of directions exist for extension of this work. The most direct extensions would be to perform experiments using more of Wikipedia, particularly for the feature exploration.

As well as the simple extensions of current features that are described here, extra data may enable the use of more complex features. For example, a feature to encode the presence of one attribute and the absence of another.

Now that we have a range of different algorithms for model estimation it may be possible to perform co-training style experiments. Even a simpler method, such as using the set of weights found by one algorithm as the initial weights for another, may lead to improved results. Additionally, the current system takes the weights produced by the perceptron algorithms, normalises them and treats them as a probability distribution in the same way as the weights from GIS and BFGS are treated. It would be interesting to explore the possibility of multi-tagging with a perceptron instead. The perceptron based algorithms can also be adjusted at run time, making it feasible to learn continuously.

Here we have presented results for training on automatically labelled Wikipedia text, but we could perform the same experiments on effectively any corpus. It would be interesting to explore the ability of the system to adapt to new domains through semi-supervised training.

7 Conclusion

This work has shown that semi-supervised supertagger training can boost parsing speed considerably and demonstrated that perceptron based algorithms can effectively estimate supertagger model parameters. To achieve this we adjusted the C&C parser's handling of backward composition, parallelised the supertagger training process, and implemented the MIRA and AP algorithms for feature weight estimation.

The change in backward composition handling provided a 50% speed boost and a further 30% was gained for parsing Wikipedia by using parsed Wikipedia as extra training data. As more Wikipedia data was used speed on the WSJ fell below the baseline, indicating that domain adaptation was occurring.

Models trained using perceptron based algorithms performed just as well, but were trained two orders of magnitude faster. Extending the feature set led to small but statistically significant improvements, including two models that achieve an F-score of 84.17% for labelled dependencies on Section 00 of the WSJ.

Initially the system produced an F-score of 83.22% on Section 00 of the WSJ, could parse the WSJ and Wikipedia at 31.7 and 30.8 sentences per second respectively, and took two hours to train the supertagging model, using only forty thousand sentences for training. Our changes enabled the construction of a model in under four minutes that achieves an F-score of 83.79 on WSJ, and speeds of 62.8 and 69.7 sentences per second for WSJ and Wikipedia respectively, ie. 2.0 times faster for WSJ, and 2.3 times faster for Wikipedia.

8 Acknowledgements

We thank the reviewers for helpful feedback. This work was supported by the Capital Markets Cooperative Research Centre Limited and a University of Sydney Merit Scholarship and aspects of the work were completed at the Summer Research Workshop on Machine Learning for Language Engineering at the Center for Language and Speech Processing, Johns Hopkins University.

References

S. Bird, E. Loper, and E. Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

C. G. Broyden. 1970. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90.

R. Chandrasekar and S. Bangalore. 1997a. Gleaning information from the web: Using syntax to filter out irrelevant information. In *World Wide Web, Stanford University*.

R. Chandrasekar and S. Bangalore. 1997b. Using supertags in document filtering: The effect of increased context on information retrieval effectiveness. In *Proceedings of RANLP '97*.

J. Chen, S. Bangalore, and V. K. Shanker. 1999. New models for improving supertag disambiguation. In *Proceedings of the 9th Meeting of EACL*, pages 188–195, Bergen, Norway, June.

J. Chen, S. Bangalore, M. Collins, and O. Rambow. 2002. Reranking an n-gram supertagger. In *Proceedings of the TAG+ Workshop*, pages 259–268, Venice, Italy, May.

N. Chinchor. 1992. Statistical significance of muc-6 results.

S. Clark and J. R. Curran. 2003. Log-linear models for wide-coverage ccg parsing. In *Proceedings of the Conf. on EMNLP*, pages 97–104. ACL.

S. Clark and J. R. Curran. 2007a. Perceptron training for a wide-coverage lexicalized-grammar parser. In *ACL 2007 Workshop on Deep Linguistic Processing*, pages 9–16, Prague, Czech Republic, June. ACL.

S. Clark and J. R. Curran. 2007b. Wide-coverage eficient statistical parsing with ccg and log-linear models. *Comp. Ling.*, 33(4):493–552.

S. Clark, J. Curran, and M. Osborne. 2003. Boostrapping pos-taggers using unlabelled data. In *Proceedings of CoNLL*.

S. Clark, A. Copestake, J. R. Curran, Y. Zhang, A. Herbelot, J. Haggerty, B. Ahn, C. Van Wyk, J. Roesner, J. K. Kummerfeld, and T. Dawborn. 2009. Large-scale syntactic processing: Parsing the web. Technical report, JHU CLSP Workshop.

S. Clark. 2002. Supertagging for combinatory categorial grammar. In *Proceedings of TAG+6*, pages 19–24, Venice, Italy, May.

M. Collins and B. Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Meeting of the ACL*, pages 111–118, Barcelona, Spain.

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conf. on EMNLP*, pages 1–8. ACL, July.

N. Cooper. 2007. Improved statistical models for supertagging.

K. Crammer and Y. Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

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

J. Eisner. 1996. Efficient normal-form parsing for Combinatory Categorial Grammar. In *Proceedings of the 34th Meeting of the ACL*, pages 79–86, Santa Cruz, CA.

R. Fletcher. 1970. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322,.

Y. Freund and R. E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

D. Goldfarb. 1970. A family of variable metric updates derived by variational means. *Mathematics of Computation*, 24:23–26.

J. Hockenmaier. 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

A. K. Joshi and S. Bangalore. 1994. Disambiguation of super parts of speech (or supertags): almost parsing. In *Proceedings of the 15th Conf. on Computational Linguistics*, pages 154–160, Kyoto, Japan, August.

T. Kasami. 1967. An efficient recognition and syntax analysis algorithm for context-free languages. Technical Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford, MA.

T. Kiss and J. Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Comp. Ling.*, 32(4):485–525.

D. McClosky, E. Charniak, and M. Johnson. 2006. Effective self-training for parsing. In *Proceedings of the NAACL Conf.*

F. Rosenblatt. 1958. The perceptron - a probabilistic model for information - storage and organization in the brain. *Psychological Review*, 65(6):386–408.

A. Sarkar, F. Xia, and A. K. Joshi. 2000. Some experiments on indicators of parsing complexity for lexicalized grammars. In *Proceedings of COLING*, pages 37–42.

A. Sarkar. 2001. Applying co-training methods to statistical parsing. In *Proceedings of the NAACL conference*, pages 1–8, Pittsburgh, Pennsylvania. ACL.

A. Sarkar, 2007. *Combining Supertagging and Lexicalized Tree-Adjoining Grammar Parsing*. MIT Press.

D. F. Shanno. 1970. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24:647–656.

M. Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Meeting of the ACL*, pages 189–196. ACL.

D. Younger. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208.

Y. Zhang, T. Matsuzaki, and J. Tsujii. 2009. Hpsg supertagging: A sequence labeling view. In *Proceedings of the 11th IWPT*, pages 210–213, Paris, France, October. ACL.

CCG parsing with one syntactic structure per n-gram

Tim Dawborn and **James R. Curran**
School of Information Technologies
University of Sydney
NSW, 2006, Australia
{tdaw3088,james}@it.usyd.edu.au

Abstract

There is an inherent redundancy in natural languages whereby certain common phrases (or n-grams) appear frequently in general sentences, each time with the same syntactic analysis. We explore the idea of exploiting this redundancy by pre-constructing the parse structures for these frequent n-grams. When parsing sentences in the future, the parser does not have to re-derive the parse structure for these n-grams when they occur. Instead, their pre-constructed analysis can be reused. By generating these pre-constructed databases over WSJ sections 02 to 21 and evaluating on section 00, a preliminary result of no significant change in F-score nor parse time was observed.

1 Introduction

Natural language parsing is the task of assigning syntactic structure to text. Initial parsing research mostly relied on manually constructed grammars. Statistical parsers have been able to achieve high accuracy since the creation of the Penn Treebank (Marcus et al., 1993); a corpus of Wall Street Journal text used for training. Statistical parsers are typically inefficient, parsing only a few sentences per second on standard hardware (Kaplan et al., 2004). There has been substantial progress on addressing this issue over the last few years. Clark and Curran (2004) presented a statistical CCG parser, C&C, which was an order of magnitude faster than those analysed in Kaplan et al. (2004). However the C&C parser is still limited to around 25 sentences per second.

This paper investigates whether the speed of statistical parsers can be improved using a novel form of caching. Currently, parsers treat each sentence independently, despite the fact that some phrases are constantly reused. We propose to store analyses for common phrases, instead of re-computing their syntactic structure each time the parser encounters them.

Our first idea was to store a single, spanning analysis for frequent n-grams. However, the most frequent n-grams often did not form constituents. Given that n-gram distributions are very long-tailed, this meant that the constituent n-grams covered only a small percentage of n-grams in the corpus.

We then turned our attention to the n-grams that were not forming constituents. First, we found that some actually *should* form constituents. However, the structure of noun-phrases in the Penn Treebank is underspecified, leading to incorrect derivations in the C&C parser's training corpus (Hockenmaier, 2003). Secondly, we investigated whether the spurious ambiguity of CCG derivations could be exploited to force frequent n-grams to compose into constituents, while still producing a semantically equivalent derivation. Here we encountered problems using the composition rule to create new constituents. Some of these problems were due to further issues with the analyses in the corpus, while others were due to the ambiguity of the n-grams.

The sparsity of n-grams in a corpus of this size meant that we had very few caching candidates to work with. Our approach may be more successful when the caching process is performed using a data set.

2 Background

We are interested in storing the parse structure for common n-grams, so that the analysis can be reused across multiple sentences. In a way, this is an extension of an important innovation in parsing: the CKY chart parsing algorithm (Younger, 1967). Our proposal is an attempt to memoise sections of the chart across multiple sentences.

Most constituency parsers use some form of chart for constructing a derivation, so our investigation could have begun with a number of different parsers. We decided to use the C&C parser (Clark and Curran, 2007) for the following reasons. First, the aim of the caching we are proposing is to improve the speed of a parser. It makes sense to look at a parser that has already been optimised, to ensure that we do not demonstrate an improvement that could have been achieved using a much simpler solution. Secondly, there are aspects of the parser's grammar formalism, Combinatory Categorial Grammar, that are relevant to the issues we want to consider.

2.1 Chart Parsing

The chart is a triangular hierarchical structure used for storing the nodes in a parse tree, as seen in Figure 1. A chart for a sentence consisting of n tokens contains $\frac{n(n+1)}{2}$ *cells*, represented as squares in the figure. Each cell in the chart contains the parse of a contiguous *span* or sequence of tokens of the sentence. As such, a cell stores the root nodes of all possible parse trees for the tokens which the cell covers. This coverage is called the *yield* of that node. This is illustrated as the linked-list style data structure highlighted as being the contents of cell $(1, 3)$ in Figure 1. The cell (p, s) in the chart contains all possible parses for all of the tokens in the range $[p, p+s)$ for a given sentence. The chart is built from the bottom up, starting with constituents spanning a single token, and then increasing the span to cover more tokens, until the whole sentence is covered.

Combinatory Categorial Grammar (CCG) (Steedman, 2000) is a lexicalised grammar formalism. This means that each word in a sentence is assigned a composite object that reflects its function in the derivation. In CCG, these objects are called lexical categories.

Categories can be built recursively from atomic

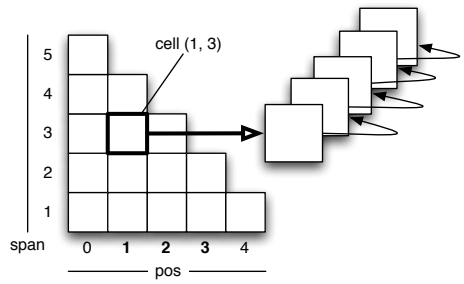

Figure 1: An illustration of the chart data structure used in parsing algorithms such as CKY

units, such as S (sentence), N (noun), and NP (noun phrase). Recursive construction of categories means that very few atomic units need to be used. For instance, there is no atomic category for a determiner in CCG. Instead, a determiner is a function which maps from a noun to a noun phrase.

Similarly, verbs are functions from some set of arguments to a complete sentence. For example, the transitive verb `like` would be assigned the category $(S\backslash NP)/NP$. Here, the slashes indicate the directionality of arguments, stating that an NP object is expected to the right, and an NP subject is expected to the left. An example CCG derivation containing the transitive verb `like` is:

$$\dfrac{\dfrac{\dfrac{\texttt{I}}{NP} \quad \dfrac{\texttt{like}}{(S[dcl]\backslash NP)/NP} \quad \dfrac{\dfrac{\texttt{the}}{NP[nb]/N} \quad \dfrac{\texttt{cat}}{N}}{NP[nb]}>}{S[dcl]\backslash NP}<}{S[dcl]}<$$

This derivation uses the rules of forward and backward application to build the representation of the sentence. Most of the information is contained in the lexical categories.

2.2 The C&C Parser

The C&C parser makes use of this property of CCG by dividing the parsing problem into two phases, following (Bangalore and Joshi, 1999). First, a *supertagger* proposes a set of likely categories for each token in the sentence. The parser then attempts to build a spanning analysis from the proposed categories, using the modified CKY algorithm described in Steedman (2000). The supertagging phase dramatically reduces the search space the parser must

explore, making the C&C parser very efficient.

3 Motivation

It is important to note that the concepts motivating this paper could be applied to any grammar formalism. However, our experiments were conducted using CCG and the C&C parser for a number of reasons, which are outlined throughout the paper.

In order for our "one structure per n-gram" idea to work in practice, the parsed data must possess two properties. Firstly, there must be a small number of n-grams which account for a large percentage of the total n-grams in the corpus. If this property was not present, this this would imply that most of the n-grams within the text appear very infrequently. As a result, the size of the database containing the memoised analyses would grow in size, as there are no n-grams which clearly are more useful to memoise than others. The result of this would be that the time taken to load the analyses from the database would exceed the time taken to let the parser construct a derivation from scratch.

The second property is that the most frequent n-grams in the corpus must have on average very few distinct analyses. If the most frequent n-grams in the corpus all occurred with a large number of different analyses, then every time we see these frequent n-grams in the future, these multiple analyses will all have to be loaded up from the database. This again would result in more time taken in the database loading than letting the parser construct the derivation from scratch. If the most frequent n-grams in the corpus thus only occur with a very small number of analyses, then the time taken to load the pre-constructed structures should be less than the time the parser will take to construct the derivation from scratch.

4 Analysis

By analysing all of the n-grams within sections 02 to 21 of CCGbank for varying n, we were able to show that, under a very basic analysis, CCGbank satisfies both the properties discussed in Section 3. The results of this analysis can be seen in Table 1.

One interesting result here is the average number of derivations varying n-grams occur with. On its first attempt at parsing a sentence, the C&C parser assigns on average 1.27 CCG categories per word (Clark and Curran, 2007). Since the average number of derivations for varying sized n-grams is less than the ambiguity introduced during the first attempt at the parsing process, this process of inserting pre-built chart structures can potentially decrease the overall parsing time, as the pre-built structures would introduce less ambiguity to the overall parse compared to what the parser would provide normally.

n-gram size	2	3	4
Avg number derivations	1.19	1.09	1.04
Always form constituents	23%	10%	5%
Never form a constituent	73%	89%	93%

Table 1: Statistics about varying sized n-gram in CCGbank sections 02 to 21

bigram	# No	# Yes	# Uniq
the company	8	1157	1
a share	3	1082	7
New York	4	868	7
a year	34	572	9
do n't	0	474	9
the market	37	410	1
did n't	0	378	11
is n't	1	367	21
The company	0	359	1
does n't	0	328	10

Table 2: Constituent statistics about the 10 most frequent bigrams in CCGbank 02 to 21 which form constituents the majority of the time

4.1 Constituents

The first idea explored is how well can we do by storing only n-grams which primarily form constituents. Table 2 shows the 10 most frequent bigrams in CCGbank sections 02 to 21 which primarily form constituents. The columns show the number of times the n-gram was seen not forming and not forming a constituent, as well as the number of unique constituent-forming analyses formed.

A number of interesting observations can be made here. Firstly, the number of times these bigrams occur drops off very quickly, with the 4th most frequent bigram appearing just under half the number

of times the most frequent bigram occurs. This drop off contradicts our first desirable property for the corpus, that there should be a large number of frequent n-grams.

Observing the numbers in the last column of Table 2, it is easily seen that only 3 of the top 10 bigrams occur with less than 5 unique derivations, which goes against our second desirable property, that the most frequent n-grams occur with very few unique derivations.

These two factors indicate that an approach which persists only constituent-forming n-grams in these databases will not perform well, as neither of the two properties discussed in Section 3 are fulfilled.

4.2 Non-constituents

Section 4.1 showed that an approach to this problem which utilises only constituent-forming n-grams most likely will not produce the desired speed boost due to the properties mentioned in Section 3 not being upheld.

The next natural direction to take is to the storing of analyses for the non-constituent-forming n-grams. The use of CCG *type raising* and *composition* allow us to store non-constituent-forming analyses in these databases for n-grams, yet still be able to use these derivations later on to form correct semantically correct spanning analyses. For example, one of the frequent non-constituent-forming occurrences of the phrase `of the` in CCGbank is

$$\dfrac{\dfrac{\texttt{of} \quad \dfrac{\texttt{the} \quad \texttt{company}}{NP/N \quad N}>}{(NP\backslash NP)/NP}}{NP\backslash NP}>$$

`The` is forward applied to `company` before `of` can be joined with `the`. Instead, we could construct the following derivation and insert it into the pre-constructed database

$$\dfrac{\texttt{of} \quad \texttt{the}}{\dfrac{(NP\backslash NP)/NP \quad NP/N}{(NP\backslash NP)/N}>\mathbf{B}}$$

Here we use CCG forwards composition to combine `of` and `the` into a constituent-forming analysis. This chart structure could be reused with CCG forwards application to construct a span of the original phrase in the following manner

$$\dfrac{\dfrac{\texttt{of} \quad \texttt{the} \quad \texttt{company}}{(NP\backslash NP)/NP \quad NP/N \quad N}}{\dfrac{(NP\backslash NP)/N}{NP\backslash NP}>\mathbf{B}}>$$

Using the forward composed version of the bigram `of the`, an analysis for the whole phrase was still able to be constructed, even though in the original derivation `of` and `the` did not form a constituent. This technique of utilising CCG forward composition and type raising allows us to add these n-grams which primarily do not form constituents, into the database.

4.2.1 Prepositional Phrase Attachment

This technique does not work all of the time, however it does work for many cases. One situation where this technique does not work is is with prepositional phrase attachment. The correct CCG derivation for the phrase `on the king of England` is

$$\dfrac{\dfrac{\dfrac{\text{X} \quad \dfrac{\texttt{on} \quad \dfrac{\texttt{the king} \quad \texttt{of England}}{NP[nb] \quad NP\backslash NP}<}{(NP\backslash NP)/NP \quad NP}>}{NP}}{NP\backslash NP}>}{NP}<$$

If we were to use in this example the same forwards composed derivation of the bigram `of the` as described earlier for the bigram `on the`, the wrong analysis would be constructed.

$$\dfrac{\dfrac{\dfrac{\text{X} \quad \dfrac{\texttt{on} \quad \texttt{the}}{(NP\backslash NP)/N \quad N}>}{NP\backslash NP}<}{NP}}{NP}< \quad \dfrac{\texttt{king} \quad \texttt{of England}}{N \quad NP\backslash NP}$$

While an NP was still the resultant overall category assigned to the phrase, the internal noun phrases are incorrect; the named entity `the king of England` is not represented within this incorrect derivation.

From an implementation point of view, being able to construct and use this forward composed parse structure for `of the` involves violating one of the normal-form constraints proposed in Eisner (1996) to eliminate CCG's "spurious ambiguity". The constraint which was violated states that the left child

of forward application cannot be the result of forward composition, as is the case in our previous example. The C&C parser implements these Eisner constraints, and as such a special rule was added to the parser to allow any chart structures which were loaded from a pre-constructed database to violate the Eisner constraints.

4.2.2 Coordination

In CCG parsing, commas can be parsed in one of two ways depending on their semantic role in the sentence. They are either used for coordination or they are absorbed. Consider the CCG derivation for the sentence shown in Figure 2. The second comma between `England` and `owned` is absorbed, as shown in the second last line of the derivation. The first comma, however, between `George` and `the king of England`, is used to express apposition. Apposition in CCG is represented using the same coordination structure which `and` uses; the conjoining combinator. This combinator is denoted as $conj$ or $\mathbf{\Phi}$ in CCG. The type signature of this combinator is

$$X \ conj \ X' \Rightarrow_{\mathbf{\Phi}} X''$$

stating that the CCG category has to be the same on both sides of a $conj$, and when the functor is invoked, the resultant category is the same. Our n-gram pre-construction attempts to memoise analyses based purely on the tokens of n-grams. Because comma appears as a $conj$, we are unable to use any n-grams which contain commas in our database, as at the token level it is not possible to determine if the comma will be absorbed or will be used in apposition.

4.3 Statistics

Table 3 shows the 15 most frequent bigrams in CCGbank sections 02 to 21. The first thing to note about this table is that only two of the top 15 most frequent bigrams primarily form a constituent, again leading to a conclusion that using only constituent-forming bigrams is not the correct approach to the problem. The second point to observe is that seven out of these 15 bigrams contain a comma, which as described in Section 4.2.2, implies these cannot be used in our database.

The Σ column shows an accumulative sum of the number of tokens covered in sections 00 to 21 just by using the bigrams in the table. The coverage figures shown in the neighbouring column show this sum as a percentage of the total number of tokens in sections 00 to 21. This shows that by considering just the 15 most frequent bigrams, a coverage of 6.5% of the total number of tokens has been achieved. If a trend like this continues linearly down this list of frequency sorted bigrams and a pre-constructed analysis for the first 1000 bigrams could be memoised, for example, there is a great potential for the parse time to be improved.

5 Evaluation

The effect of these n-gram databases on the parsing process is evaluated in terms of the overall parsing time, as well as the accuracy of the resultant derivations. The accuracy is measured in terms of F-score values for both labelled and unlabelled dependencies when evaluated against the predicate-argument dependencies in CCGbank (Clark and Hockenmaier, 2002). The parsing times reported do not include the time to load the grammar, statistical models, or our database.

6 Implementation

6.1 Data

The models used by the C&C parser for our experiments were trained using two different corpora. The WSJ models were trained using the CCG version of the Penn Treebank, CCGbank (Hockenmaier, 2003; Hockenmaier and Steedman, 2007), which is available from the Linguistic Data Consortium[1]. The second corpus is a version of CCGbank where the noun phrase bracketing has been corrected (Vadas and Curran, 2008; Vadas, 2009).

6.2 Tokyo Cabinet

Tokyo Cabinet[2] is an open source, lightweight database API which provides a number of different database implementations, including a hash database, B+ tree, and a fixed-length key database. Our experiments used Tokyo Cabinet to store the pre-constructed n-grams because of its ease of use, speed, and maximum database size (8EB). A large

[1] `http://ldc.upenn.edu/`

[2] `http://tokyocabinet.sourceforge.net/`

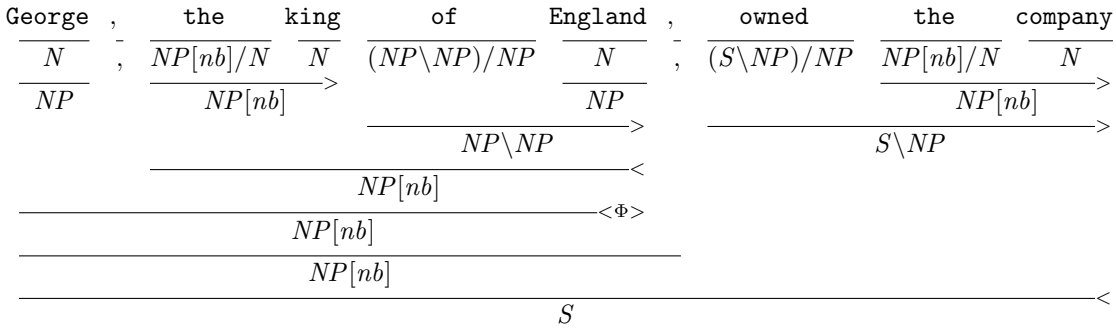

Figure 2: A CCG derivation containing commas used for apposition and absorption

bigram	Constituent			Coverage		Ambiguity
	# No	# Yes	# Uniq	Σ	%	
of the	4936	0	0	9872	1.06	1.031
in the	3911	5	1	17704	1.90	1.747
, the	3489	0	0	24682	2.66	1.005
, and	2219	9	3	29138	3.13	1.000
, a	2167	0	0	33472	3.60	1.000
, which	1705	0	0	36882	3.97	1.118
for the	1638	0	0	40158	4.32	1.958
to the	1588	1	1	43336	4.66	1.925
on the	1533	0	0	46402	4.99	1.962
, said	1258	0	0	48918	5.26	1.290
, but	1193	1	1	51306	5.52	1.045
the company	8	1157	1	53636	5.77	1.000
, he	1165	0	0	55966	6.02	1.000
that the	1150	0	0	58266	6.27	1.283
a share	3	1082	7	60436	6.50	1.107

Table 3: Constituent statistics about the 15 most frequent bigrams in CCGbank 02 to 21. The columns show the number of times the bigram was seen forming a non-constituent, forming a constituent, and then the number of unique constituent-forming chart structures. The next two columns show accumulatively what percentage of sections 02 to 21 these bigrams alone cover. The last column shows the ambiguity the C&C supertagger associates to each n-gram

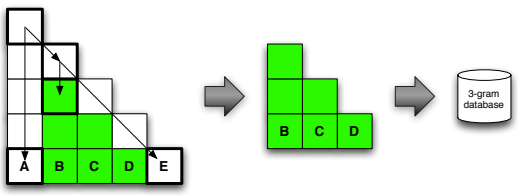

Figure 3: When creating the trigram database, if a trigram forms a constituent in the chart, it is added to the database

maximum database size is important because more data is better for the database construction phase.

6.3 Constructing the n-gram Databases

The construction of the final database is a multi-stage process, with intermediate databases being generated and then refined. The first stage in this process is to parse all of the training data, which in our case is WSJ sections 02 to 21. The parse tree for every sentence is then analysed for constituent-forming n-grams. If a constituent-forming n-gram is found and its size (number of tokens) is one for which we would like to construct a database for, then the n-gram and its corresponding chart structure are written out to a database. These first stage databases are implemented using a simple key-value Tokyo Cabinet hash database. The structure of the keys and values in this database are

Key = `(n-gram, hash of chart)`
Value = `(chart, occurrence counter)`

The `chart` attribute in the value is a serialised version of the chart which can be unserialised at some later point for reuse. The `occurrence` counter is incremented each time an occurrence of a key is seen in the parsed training data. A record is also kept in the database for the number of times a particular n-gram was seen forming a non-constituent, for use in the filtering stage discussed in later Section 6.4.

This process of n-gram chart serialisation is illustrated in Figure 3. When parsing the sentence `A B C D E`, the trigram `B C D` formed a constituent in the spanning analysis for the sentence. Because it formed a constituent, the trigram is added to the first stage trigram database.

6.4 Frequency Reduction

When constructing the initial set of databases over a body of text, a large number of the n-grams which were memoised should not be kept in the final databases because they occur too infrequently, or because the number of times they are seen forming a non-constituent outweighs the number of times they are seen forming a constituent. As such, a frequency based filtering stage is performed on the initial set of databases to produce the final database.

$$\frac{C_0}{\sum_{k \neq 0} C_k} < X \tag{1}$$

$$m = \arg\max_k C_k \tag{2}$$

$$\frac{(\sum_k C_k) - m}{m} < X \wedge m > Y \tag{3}$$

An n-gram was chosen to be filtered out differently depending on whether or not it was seen forming a non-constituent during the database development phase. Equations 1 and 3 describe the predicates which need to be fulfilled in order for a particular n-gram not to be filtered out. In these inequalities, C is a mapping from chart structure to frequency count for the current n-gram, the 0th index into C is the non-constituent-forming frequency count, and X and Y are parameters to the filtering process.

If an n-gram was seen forming a non-constituent during the initial database development phase, then Equation 1 is used. If an n-gram was never seen forming a non-constituent during the development phase, then Equation 3 is used.

The values given to the X and Y parameters in the filtering process were determined through a trial and error process, training on sections 02 to 21 and testing on section 00 of the noun phrase corrected CCGbank. For all of our results, X was set to 0.05 and Y was set to 15.

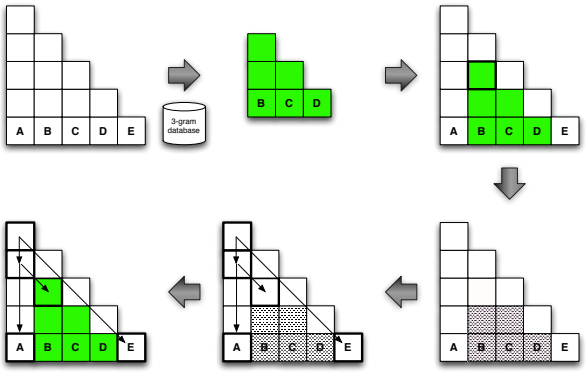

Figure 4: Illustration of using the n-gram databases. The trigram `B C D` is loaded from the pre-constructed database, and blocks out the corresponding cells

6.5 Using the n-gram Databases

Once the n-gram database has been constructed, it is used when parsing sentences in the future. For every sentence that is parsed, the parser checks to see if any n-gram contained within the current sentence exists within the database, and if so, uses the memoised analysis for the n-gram. This process is illustrated in Figure 4.

This n-gram check is performed by iterating top to bottom, left to right through the chart for the current sentence. A consequence of this is that if two n-grams overlap and both exist in the database, then only the first n-gram encountered will have its analyses loaded in from the database. Once the analyses are loaded into the current chart for the n-gram, the corresponding cells in the current chart are blocked off from further use in the parse tree creation process (CKY), as illustrated in Figure 4. It is due to this cell blocking that the pre-constructed charts for the 2nd overlapping n-gram are not also loaded.

Model		Baseline	2-gram	3-gram
WSJ derivs	Time	82.8	82.8	82.6
	LF	85.26	85.26	85.26
	UF	92.01	92.01	92.01
	Cov	98.64	98.64	98.64
WSJ hybrid	Time	84.1	83.8	83.9
	LF	87.40	87.40	87.40
	UF	93.10	93.10	93.10
	Cov	98.64	98.64	98.64
NP derivs	Time	84.1	84.0	84.3
	LF	83.60	83.60	83.60
	UF	90.48	90.48	90.48
	Cov	98.54	98.54	98.54
NP hybrid	Time	84.8	84.9	85.3
	LF	85.88	85.88	85.88
	UF	91.63	91.63	91.63
	Cov	98.54	98.54	98.54

Table 4: The speed versus performance trade-off for varying sized n-grams evaluated on CCGbank 00 using different parsing models. The evaluation attributes are parse time (s), labelled and unlabelled F-score (%), and percentage of sentences covered

7 Results

A set of experiments were conducted using CCGbank sections 02 to 21 as the corpus for developing our database. This corpus was parsed using a variety of statistical parsing models. Section 00 was then used for evaluation. Table 4 shows our preliminary results. The first two parsing models used were trained on the original CCGbank (WSJ derivs and hybrid), and the second two models were trained on the noun phrase corrected CCGbank corpus described in Vadas and Curran (2008) (NP derivs and hybrid). The databases used to obtain these results contained only constituent-forming n-grams.

These results show a non-significant change in speed nor F-score. One positive aspect of this non-significant change is that performance did not decrease even though additional computation is needed to perform our database lookups and chart insertion. The C&C parser is already very fast, and the amount of time taken to perform the chart loading and insertion from the databases happens to be very similar to the time taken to construct the derivations from scratch.

Another experiment was then performed in order to assess the potential of using non-constituent-forming n-grams for memoisation. The bigrams `of the` and `in the` are the two most frequently occurring non-constituent-forming bigrams in CCGbank sections 02 to 21. In order to assess the viability of using non-constituents in our database, our experiments here used only the most frequently occurring analyses for these two bigrams. If no improvement in performance is observed using the most frequently occurring bigrams, then the idea is not worth pursuing further.

	Baseline	`of the`	`in the`	Combined
Time	67.2	67.0	65.8	66.0
LF	87.58	87.30	87.30	87.24
UF	93.14	92.86	92.89	82.83
Cov	94.30	94.30	84.41	84.30

Table 5: Memoised structures were constructed for the most frequent derivations for varying non-constituent-forming bigrams, which were then used and evaluated against section 00 of the noun-phrase corrected CCGbank

The results of these experiments can be seen in Table 5. As was the case in our constituent-forming experiment, no significant change in performance was achieved; positive or negative.

8 Conclusion

Through the analysis of this one structure per n-gram idea using CCG, combined with a preliminary set of empirical results, we have shown that memoising parse structures based on frequently occurring n-grams does not result in any form of performance improvement.

9 Acknowledgments

We would like to thank the anonymous reviewers for their useful feedback. This work was partially supported by the Capital Markets Cooperative Research Centre Limited. Aspects of this work were carried out as part of the Summer Research Workshop on Machine Learning for Language Engineering at the Center for Language and Speech Processing, Johns Hopkins University.

References

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: an approach to almost parsing. *Compu-*

tational Linguistics, 25(2):237–265.

Stephen Clark and James R. Curran. 2004. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics, Main Volume*, pages 103–110, Barcelona, Spain, July.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Stephen Clark and Julia Hockenmaier. 2002. Evaluating a wide-coverage CCG parser. In *Proceedings of the LREC Workshop on Beyond Parseval*, pages 60–66, Las Palmas, Spain.

Jason Eisner. 1996. Efficient normal-form parsing for combinatory categorical grammar. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 79–86, Morristown, NJ, USA. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2007. CCG-Bank: A corpus of CCG derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396.

Julia Hockenmaier. 2003. *Data and Models for Statistical Parsing with Combinatory Categorical Grammar*. Ph.D. thesis, University of Edinburgh.

Ronald M. Kaplan, Stefan Riezler, Tracy H. King, John T. Maxwell III, and Er Vasserman. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proceedings of HLT-NAACL04*, pages 97–104, Boston, Massachusetts, USA, May. Association for Computational Linguistics.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA.

David Vadas and James R. Curran. 2008. Parsing noun phrase structure with CCG. In *Proceedings of the 46th Meeting of the Association for Computational Linguists: HLT*, pages 335–343, Columbus, Ohio, June. Association for Computational Linguistics.

David Vadas. 2009. *Statistical Parsing of Noun Phrase Structure*. Ph.D. thesis, University of Sydney.

D Younger. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208.

An Update on PENG Light

Colin White
Department of Computing
Macquarie University
Sydney NSW 2109, Australia
colcwhite@gmail.com

Rolf Schwitter
Centre for Language Technology
Macquarie University
Sydney NSW 2109, Australia
Rolf.Schwitter@mq.edu.au

Abstract

This paper presents an update on PENG Light, a lightweight and portable controlled natural language processor that can be used to translate a well-defined subset of English unambiguously into a formal target language. We illustrate by example of a Firefox extension that provides a simple interface to the controlled natural language processor how web pages can be annotated with textual information written in controlled natural language and how these annotations can be translated incrementally into first-order logic. We focus in particular on technical aspects of the controlled language processor and show in detail how look-ahead information that can be used to guide the writing process of the author is generated during the parsing process. Additionally, we discuss what kind of user interaction is required for processing unknown content words.

1 Introduction

Computer-processable controlled natural languages are well-defined and tractable subsets of natural languages that have been carefully designed to avoid constructions that may cause ambiguities (Fuchs et al., 1998; Schwitter, 2002; Sowa, 2004; Barker et al., 2007). Instead of encoding a piece of knowledge in a formal language that is difficult to understand for humans, a controlled natural language can be used to express the same information in a direct way using the vocabulary of the application domain. There is no need to formally encode this information since a computer-processable controlled natural language can be translated automatically and unambiguously into a formal target language by a machine. This has the advantage that everybody who knows English can understand a text written in controlled natural language and that a machine can process this text since it corresponds to a formal notation. In order to support the writing of these texts, text- and menu-based predictive editing techniques have been suggested that guide the writing process of the author (Tennant et al., 1983; Schwitter et al., 2003; Thompson et al., 2005; Kuhn, 2008). These techniques give the author a way to match what he or she wants to express with the processing capabilities of the machine and result in user-friendly and self-explanatory interfaces. However, practically no details have been published so far how these predictive techniques can actually be implemented in a controlled natural language processor that processes a text incrementally. In this paper, we will make up for this neglect and show how these predictive techniques have been implemented for the controlled natural language processor of PENG Light and discuss how the language processor uses these techniques while communicating over an HTTP connection with a simple AJAX-based tool designed for annotating web pages in controlled natural language.

The rest of this paper is structured as follows: In Section 2, we give a brief overview of existing computer-processable controlled natural languages. In Section 3, we introduce FoxPENG, a simple Firefox extension that we have built for annotating web pages with controlled natu-

ral language and use this extension as a vehicle for motivating predictive editing techniques. In Section 4, we give a brief introduction to the controlled natural language PENG Light and show how sentences can be anaphorically linked and finally translated into discourse representation structures. In Section 5, we present the latest version of the chart parser of PENG Light and focus on the incremental processing of simple and compound words. In Section 6, we discuss what user interaction is required for processing unknown content words and suggest a linear microformat for specifying feature structures. In Section 7, we summarise the advantages of our controlled natural language approach for specifying a piece of knowledge.

2 Related Controlled Natural Languages

During the last decade, a number of computer-processable controlled natural languages have been designed and used for specification purposes, knowledge acquisition and knowledge representation, and as interface languages to the Semantic Web – among them Attempto Controlled English (ACE) (Fuchs et al., 1998; Fuchs et al., 2008), Processable English (Schwitter, 2002), Common Logic Controlled English (Sowa, 2004), and recently Boeing's Computer-Processable Language (Clark et al., 2005; Clark et al., 2007). Some machine-oriented controlled natural languages require the author to learn a small number of construction and interpretation rules (Fuchs et al., 2008), while other controlled natural languages provide writing support which takes most of the burden of learning and remembering the language from the author (Thompson et al., 2005). The commercial success of the human-oriented controlled natural language ASD Simplified Technical English (ASD, 2007) suggests that people can learn to work with restricted English and that good authoring tools can drastically reduce the learning curve of the language.

The language processors of ACE and PENG Light are both based on grammars that are written in a definite clause grammar (DCG) notation (Pereira and Shieber, 1987). These DCGs are enhanced with feature structures and specifically designed to translate declarative and interrogative sentences into a first-order logic notation via discourse representation structures (Kamp and Reyle, 1993). In contrast to ACE that uses the DCG directly and resolves anaphoric references only after a discourse representation structure has been constructed, PENG Light transforms the DCG into a format that can be processed by a top-down chart parser and resolves anaphoric references during the parsing process while a discourse representation structure is built up.

3 The FoxPENG Toolbar

In order to annotate web pages with information that is at the same time human-readable and machine-processable, we developed FoxPENG, an AJAX-based Firefox extension (see Figure 1). FoxPENG supports the writing of annotations with the help of look-ahead information that indicates what syntactic categories or word forms can follow the current input. This look-ahead information is dynamically generated and updated by the language processor while an annotation is written. This application has some similarities to Google Suggest[1] and other autocomplete mechanisms[2] provided by source code editors, database query tools, and command line interpreters.

In contrast to Google Suggest that guesses what an author writes, the look-ahead information displayed in FoxPENG is a side-effect of the parsing process of the controlled natural language. FoxPENG communicates asynchronously with the language processor of PENG Light via a Prolog HTTP server[3] using JSON[4] (JavaScript Object Notation) as data-interchange format.

Once a connection to the Prolog server has been established, FoxPENG makes a request for the initial look-ahead categories to be displayed along the lower section of the toolbar. The author can now start typing an annotation that begins with a word form that falls under the corresponding look-ahead categories. Once a simple word form has been entered and the space bar has been pressed, an HTTP request is sent to the language processor containing the word form as well as the position of the word in the sentence.

[1] http://www.google.com/support/websearch/bin/answer.py?answer=106230
[2] http://en.wikipedia.org/wiki/Autocomplete/
[3] http://www.swi-prolog.org/packages/http.html
[4] http://json.org/

Figure 1: FoxPENG – Firefox Extension

The chart parser of PENG Light processes this information and either replies with a set of new look-ahead categories and word forms to choose from or with spelling suggestions in case of an error. The spelling suggestions are derived from the entries in the linguistic lexicon of PENG Light. If a content word is not misspelled and cannot be found in the linguistic lexicon, then the author can specify this word directly in the text area of FoxPENG using a microformat for linearised feature structures (see Section 6). An annotation can consist of more than one PENG Light sentence, and the language processor can resolve anaphoric references during the writing process using the standard accessibility constraints imposed by the discourse representation structures (see Section 4). The resulting discourse representation structure can be further translated into the input format of an automated reasoning engine and then be used for various reasoning tasks, among them for question answering. We have used the theorem prover and model builder E-KRHyper (Baumgartner et al., 2007) as reasoning service for PENG Light.

The author can publish a FoxPENG annotation as part of an RSS feed that contains a link to the annotated web page. In principle, any RSS feed aggregator can subscribe to such an RSS feed that is written in controlled natural language. This has the benefit that annotations are not only human-readable but also machine-processable (with the help of a PENG-compliant language processor).

4 PENG Light

PENG Light is a computer-processable controlled natural language that can be used for knowledge representation (Schwitter, 2008). At first glance, PENG Light **looks like** a subset of natural language, but PENG Light is actually a **formal** language since the language is designed in such a way that it can be translated unambiguously into a formal target representation. The vocabulary of PENG Light consists of predefined function words (determiners, coordinators, subordinators, prepositions and query words), a small number of predefined phrases (e.g. *there is*, *it is false that*) and content words (nouns, proper nouns, verbs, adjectives and adverbs) that can be defined by the author. A PENG text is a sequence of anaphorically interrelated sentences that consist of simple and complex sentences.

4.1 Sentences in PENG Light

PENG Light distinguishes between simple, complex, and interrogative sentences. Simple sentences consist of a subject and a verb (1-6), necessary complements (2-5), and optional adjuncts (5+6):

1. David Miller works.
2. David Miller teaches COMP249.
3. David Miller sends a letter to Mary.
4. David Miller is in the lecture hall.
5. David Miller teaches COMP249 on Monday.
6. David Miller works fast.

Complex sentences are built from simpler sentences through quantification (7+8), negation (9-11), subordination (12), and coordination (13):

7. Every professor teaches a unit.
8. David Miller teaches [exactly | at least | at most] two units.
9. If is false that a professor teaches COMP225.
10. No professor teaches COMP225.
11. David Miller is not a professor.

```
[drs([A, B, C],
  [theta(A, theme, C)#[1],
   event(A, working)#[1],
   theta(A, location, B)#[1],
   named(B, macquarie_university)#[1, [third, sg, neut],
                                    ['Macquarie','University']],
   named(C, david_miller)#[1, [third, sg, masc],['David','Miller']]])]
```

Figure 2: Annotated Discourse Representation Structure in PENG Light

12. David who teaches a unit supervises Mary.
13. David teaches COMP249 and supervises Mary.

A special form of complex sentences are conditionals (14) and definitions (15):

14. If David Miller works on Monday then Sue Rosenkrantz works on Tuesday.
15. A professor is defined as an academic who leads a research group and who teaches at least two units.

PENG Light distinguishes two types of interrogative sentences: *yes/no*-questions (16) and *wh*-questions (17):

16. Does David Miller teach a tutorial on Monday?
17. When does David Miller who convenes COMP249 teach a tutorial?

Interrogative sentences are derived from simple PENG Light sentences and serve the same purpose as queries in a formal query language.

4.2 Anaphora Resolution

In PENG Light, proper nouns, definite noun phrases and variables (that build an unambiguous alternative to pronouns) can be used anaphorically. The anaphora resolution algorithm of PENG Light resolves an anaphorically used noun phrase with the most recent accessible noun phrase antecedent that matches fully or partially with the anaphor and that agrees in person, number and gender with that anaphor. The anaphora resolution algorithm of PENG Light is embedded into the grammar and triggered whenever a noun phrase has been processed.

If a definite noun phrase can not be resolved by the anaphora resolution algorithm, it is interpreted as an indefinite noun phrase and introduces a new discourse referent into the universe of discourse.

4.3 Discourse Representation Structures

The language processor (DRS version) of the PENG Light system translates texts incrementally into TPTP notation (Sutcliffe and Suttner, 1998) with the help of discourse representation structures (DRSs) (Kamp and Reyle, 1993). The DRSs used in PENG Light rely on an event based notation (Davidson, 1967; Parsons, 1994) and a small number of thematic roles similar to (Kipper et al., 2008). Some of the conditions in the resulting DRS are annotated with syntactic information in order to improve the processability of the DRS by other system components (for example, the anaphora resolution algorithm). These conditions have the form `Pred#Anno` whereas `Pred` is a predefined predicate and `Anno` is a list that contains syntactic information. Figure 2 shows a simple DRS for the sentence (18):

18. David Miller works at Macquarie University.

The grammar of PENG Light contains not only feature structures for DRSs but also feature structures for syntactic and pragmatic information. In PENG Light, the construction of a DRS always runs in parallel with the construction of a syntax tree and a paraphrase. The paraphrase clarifies how the input has been interpreted by the grammar and can be used to show the author all relevant substitutions.

5 Chart Parsing in PENG Light

The grammar of PENG Light is specified in definite clause grammar (DCG) notation. However, the direct execution of a DCG would create many partial structures and destroy them while backtracking. This is not particularly efficient for generating look-ahead information (Kuhn and Schwitter, 2008). In order to avoid unnecessary repetition of work and to generate look-

ahead information efficiently, the DCG is transformed via term expansion (a well-known logic programming technique) into a notation that can be processed by the chart parser of PENG Light. The chart parser is based on work by (Gazdar and Mellish, 1989) but has been substantially extended to better support the incremental processing of PENG Light sentences, in particular to allow for generating look-ahead information, for processing compound words, and for resolving anaphoric references during the parsing process.

5.1 Basics of Chart Parsing

In general, a chart parser stores well-formed constituents and partial constituents in a chart (= table) that consists of a series of numbered vertices that are linked by edges (Kay, 1980). The vertices mark positions in the input and edges tell us what constituents have been recognised where for a given set of grammar rules. The chart parser of PENG Light represents edges as predicates with six arguments:

```
edge(SN,V1,V2,LHS,RHSFound,RHSToFind)
```

The first argument `SN` stores the sentence number, the subsequent two arguments state the existence of an edge between vertex `V1` and vertex `V2`, and the next three arguments represent information about the grammar rule. `LHS` is a category on the left-hand side of a grammar rule, `RHSFound` is a list of confirmed categories that have been found on the right-hand side of the grammar rule, and `RHSToFind` is a list of categories on the right-hand side that still need to be confirmed.

Inactive edges represent well-formed constituents where the right-hand side `RHSToFind` is empty, and active edges represent partial constituents where the right-hand side is not empty. The fundamental rule of chart parsing states what should happen when an inactive edge and an active edge meet. It specifies that whenever an inactive edge can extend an active edge, then a new edge (that is either active or inactive) is built and added to the chart. Finally, the prediction rule of chart parsing generates new active edges and is dependent on the first category on the right-hand side of a grammar rule and the previous state of the fundamental rule.

5.2 Initialising the Chart

We initialise the chart top-down and guarantee that a number of active edges are added to the chart using a failure-driven loop (see (Kuhn and Schwitter, 2008) for details). This initialisation process creates, for example, the following (radically simplified) set of active edges that start and end at vertex `0`for the first sentence:

```
edge(1,0,0,d(_),[],[s(_),pm(_)]).
edge(1 0,0,s(_),[],[np(_),vp(_)]).
edge(1,0,0,np(_),[],[det(_),noun(_)]).
edge(1,0,0,np(_),[],[pn(_)]).
edge(1,0,0,det(_),[],[lexicon(_)]).
edge(1,0,0,pn(_),[],[lexicon(_)]).
...
```

Additionally, the initialisation process adds the initial look-ahead information to the knowledge base. This look-ahead information consists of all those lexical categories (`lexicon(_)`) that occur as the first element in the list of unconfirmed categories and is stored in the following way in the knowledge base:

```
lookahead(FeatureStructures).
```

This makes it easy to extract the look-ahead information since the argument *`FeatureStructures`* consists of a list of feature-value pairs that contain the required syntactic and semantic information; additionally, the language processor can use this information to extract all word forms from the linguistic lexicon for a specific category that obey the current grammatical constraints.

5.3 Processing Simple Words

Once the chart has been initialised and a set of look-ahead categories has been displayed, the author can enter the first word form that belongs to one of these categories. Simple word forms are stored in the following way in the lexicon of PENG Light:

```
lexicon([
  cat:pn,
  wform:['John'],
  syn:[third, sg, masc],
  sem:[[I, person], atomic],
  con:named(I, john)]).
```

The chart parser uses the rule below together with Rule 4 in Figure 4 to look up a simple word (for example, *John*) in the lexicon:

```
(1)   add_edge(SN,V1,V2,LHS,Found,RHS) :-
        edge(SN,V1,V2,LHS,Found,RHS), !.

(2)   add_edge(SN,V1,V2,LHS,Found,[]) :-
        assert_edge(SN,V1,V2,LHS,Found,[]),
        apply_fundamental_rule(SN,V1,V2,LHS,[]).

(3)   add_edge(SN,V1,V2,LHS,Found,[RHS|RHSs]) :-
        assert_edge(SN,V1,V2,LHS,Found,[RHS|RHSs]),
        apply_fundamental_rule(SN,V1,V2,LHS,[RHS|RHSs]),
        predict_active_edges(SN,V2,RHS),
        update_lookahead_cats(SN,V2,RHS).
```

Figure 3: Add Edges

```
start_chart(SN,V1,V2,Word) :-
  foreach(word(T,SN,V1,V2,Word,LHS),
    add_edge(SN,V1,V2,LHS,Word,[])).
```

In a first step, the chart parser generates inactive edges for each occurrence of the word form in the lexicon using Rule 2 in Figure 3, after checking if such an edge does not already exist (Rule 1 in Figure 3). In the second step, the chart parser applies the fundamental rule of chart parsing recursively to inactive edges (Rule 2 in Figure 3) and active edges (Rule 3 in Figure 3). In the next step, the prediction rule of chart parsing is applied that looks for each grammar rule that has the category `RHS` previously used by the fundamental rule on the left-hand side (`LHS`), and generates new active edges for these categories. Once this has been done, the look-ahead information is updated (Rule 3 in Figure 3). For our example, this results in the following update of the chart:

```
edge(1,0,1,s(_),[np(_)],[vp(_)]).
edge(1,0,1,np(_),[pn(_)],[]).
edge(1,0,1,pn(_),['John'],[]).
edge(1,1,1,vp(_),[],[iv(_)])
edge(1,1,1,iv(_),[],[lexicon(_)])
...
```

Note that for each subsequent simple word form that the author enters, new grammar rules are triggered, new edges are added to the chart, and a new set of look-ahead categories is generated, extracted and then – in our case – sent to FoxPENG.

5.4 Processing Compound Words

As we have seen in the last section, the chart parser of PENG Light handles the input in an incremental fashion on a word by word basis. This creates problems for compound words. Each compound word is stored in the linguistic lexicon as a single entry of the following form:

```
lexicon([
  cat:noun,
  wform:[laptop,computer,bag],
  syn:[third,sg,neut],
  sem:[[I,entity],atomic],
  con:object(I,laptop_computer_bag)]).
```

This requires a special treatment of compound words by the chart parser since there are no grammar rules that describe the structure of these compound words, and a compound word can compete with other compound words or a simple word during processing. The chart parser of PENG Light uses three different rules (Rules 1-3 in Figure 4) to process compound words. The basic idea behind these rules is to retrieve each compound word only once from the linguistic lexicon as soon as the first element of a compound word becomes available and then maintain a store (`compound_word/6`) that is used to process all subsequent elements of the compound word (similar to edges). This will finally result in a single edge for the compound word in the chart.

Let us assume that the author is in the process of writing the compound noun *laptop computer bag*. After the first word (*laptop*) becomes available, the chart parser looks this word up in the linguistic lexicon using Rule 3 (and 4) in Figure 4, finds that this word is the first element of a compound word, and then checks if an active edge exists that corresponds to the category (`LHS`) on left-hand side of the grammar rule that

```
(1)   word(compound,SN,V1,V2,[Word],LHS) :-
        compound_word(SN,V0,V1,LHS,Found,[Word]),
        add_edge(SN,V0,V2,LHS,[Word|Found],[]).

(2)   word(compound,SN,V1,[Word],LHS) :-
        compound_word(SN,V0,V1,LHS,Found,[Word,LAH|LAHs]),
        edge(SN,V0,V0,LHS,[],[RHS|RHSs]),
        update_compound_word(SN,V0,V2,LHS,[Word|Found],[LAH|LAHs]),
        update_lookahead_cats(SN,V2,[LAH|LAHs]).

(3)   word(compound,SN,V1,[Word],LHS) :-
        call( LHS ==> [ lexicon([cat:Cat,wform:[Word,LAH|LAHs]|Rest],_) ] ),
        edge(SN,V1,V1,LHS,[],[RHS|RHSs]),
        call( lexicon([cat:Cat,wform:[Word,LAH|LAHs]|Rest],_) ),
        update_compound_word(SN,V1,V2,LHS,[Word],[LAH|LAHs]),
        update_lookahead_cats(SN,V2,[LAH|LAHs]).

(4)   word(simple,SN,V1,V2,[Word],LHS) :-
        \+ compound_word(SN,V0,V1,_,Found,[Word]),
        call( LHS ==> [ lexicon([cat:Cat,wform:[Word]|Rest],_) ] ),
        call( lexicon([cat:Cat,wform:[Word]|Rest],_) ).
```

Figure 4: Processing Simple and Compound Words

has been used for the lexicon lookup. In the next step, the chart parser updates the compound noun using the predicate `update_compound_word/6`. This predicate stores the sentence number (`SN`), the starting position (`V1`) and end position (`V2`) of the first element of the compound word, the category (`LHS`) on the left-hand side of the grammar rule, the found word (`[Word]`), and the remaining elements (`[LAH|LAHs]`) of the compound word. These remaining elements serve as new look-ahead information. If the author enters the next word (*computer*), the chart parser looks up this word in the store of compound words using Rule 2 in Figure 4, removes this word, checks if an active edge exists for this word, and then updates the store for compound words and the look-ahead categories. Finally, if the author enters the last element (*bag*) of the compound word, then the chart parser uses Rule 1 in Figure 4 and checks if the word is the last element of a compound noun, and adds a new edge to the chart that spans the entire compound word using Rule 2 in Figure 3, followed by a call to the fundamental rule. Note that this is a generic solution that can be used to process all categories of compound words.

6 Unknown Content Words

In principle, most function words can be displayed directly in the interface since there number is relatively small in controlled natural languages, but content words need to be structured in menus. These menus can be updated dynamically while a text is written. If the number of content words is large, then a copy of the linguistic lexicon can be loaded and maintained on the client side, and the task of the language processor is then reduced to inform the client about which categories of content words can follow the current input. In our case, PENG Light communicates with the client via a JSON object that has the following (simplified) form:

```
{"lookahead": [
    ["adj",["colour",
            "shape"]],
    ["noun",["masculine",
             "feminine",
             "masculine-feminine",
             "neuter-time",
             "neuter-entity"]] ],
 "paraphrase": ["A"] }
```

This object tells the client that either an adjective or a noun can follow the current input (in our case an indefinite determiner) and specifies

syntactic and semantic constraints for these categories. This information becomes also useful – as we will see below – if a content word is not available in the linguistic lexicon.

In real world applications, there are always cases where a required content word is missing from the linguistic lexicon. PENG Light allows the author to define a content word during the writing process. If the author enters a content word into the input field of the editor that is not yet defined in the lexicon, then the spelling corrector of PENG Light is used and provides alternative spellings that correspond to available lexical entries. PENG Light uses the Daumerau rules for this purpose that cover about 80% of human spelling errors (Damerau, 1964). These rules deal with the insertion, deletion, and substitution of single characters, and the transposition of two characters.

If a content word is not misspelled and not in the lexicon, then the author has to add the word form to the linguistic lexicon. The grammar already constrains the set of syntactic and semantic features that the author has to specify for a new content word. Let us assume that the word *laptop* is not yet in the lexicon. In this case the author has to specify only that this word is *neuter* and belongs to the sortal category *entity* but not that it is singular since this information can be derived from the current position of the word in the sentence and the information in the grammar. PENG Light accepts in-line specifications of linearised feature structures in a "microformat" notation, for example:

```
Input: A +n-n-e:laptop+
```

Note that this feature structure is an abbreviated notation that contains syntactic and semantic information derived from the feature structure provided by the JSON object. The plus symbol (+) functions as a control character that switches from the text entry mode to the vocabulary entry mode. The subsequent character sequence (*n-n-e*) represents the required feature structure, followed by a colon (:), the actual word form (*laptop*), and a plus symbol (+) that quits the vocabulary entry mode. The plus sign at the end of the word is necessary in order to deal with compound words, for example:

```
Input: A +n-n-e:laptop computer bag+
```

Using this approach, the author needs to specify only a minimal set of features and does not need to leave the text area in order to add a new content word to the user lexicon. For each category of content words, PENG Light maintains a list of unapproved words. A new content word is always checked against this list, before it is added to the user lexicon. Once a new word form has been successfully added to the linguistic lexicon, it is immediately parsed by the language processor and new look-ahead information is generated. Note that the author can only add new content words (adjectives, nouns, verbs and adverbs) using this microformat but not function words.

Similar to adding new content words, existing content words can be removed from the user lexicon. Alternatively, the microformat for feature structures is available from a menu of options.

7 Conclusions

In this paper, we presented an update on the controlled natural processor of PENG Light and showed how the language processor communicates with FoxPENG, an AJAX-based Firefox extension, using JSON as data-interchange format. For each approved word form that the author enters into the text area of this tool, the chart parser of PENG Light generates a set of look-ahead categories that determine what categories of word forms can follow the current input. This way only syntactically correct input is accepted by the language processor that can be translated unambiguously into a formal target notation. We focused in particular on an extension of the chart parser of PENG Light and showed in detail how compound words for which no grammar rules exist can be parsed incrementally during the writing process. We solved the unknown word problem with the help of a microformat for linearised feature structures that allows an author to specify unknown content words during the parsing process using minimal linguistic information. The controlled natural language PENG Light can be used as a high-level specification language for different kinds of knowledge systems and can help to solve the knowledge acquisition problem. Depending on the expressivity of the controlled language, the input can currently be translated into first-order logic or into a variant of description logic.

References

ASD 2007. ASD Simplified Technical English. *Specification ASD-STE100*, International specification for the preparation of maintenance documentation in a controlled language, Issue 4, January.

K. Barker, B. Agashe, S.-Y. Chaw, J. Fan, N. Friedland, M. Glass, J. Hobbs, E. Hovy, D. Israel, D.S. Kim, R. Mulkar-Mehta, S. Patwardhan, B. Porter, D. Tecuci, and P. Yeh. 2007. Learning by Reading: A Prototype System, Performance Baseline and Lessons Learned. In: *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 280–286.

P. Baumgartner, U. Furbach, and B. Pelzer. 2007. Hyper Tableaux with Equality. In: *Proceedings of CADE-21*, LNAI 4603, pp. 492–507.

P. Clark, P. Harrison, T. Jenkins, T. Thompson, and R. Wojcik. 2005. Acquiring and Using World Knowledge Using a Restricted Subset of English. In: *Proceedings of FLAIRS'05*, pp. 506–511.

P. Clark, P. Harrison, J. Thompson, R. Wojcik, T. Jenkins, and D. Israel. 2007. Reading to Learn: An Investigation into Language Understanding. In: *Proceedings of AAAI 2007 Spring Symposium on Machine Reading*, pp. 29–35.

F.J. Damerau. 1964. A technique for computer detection and correction of spelling errors. In: *Communications of the ACM*, 7(3), pp. 171–176.

D. Davidson. 1967. The logical form of action sentences. In: Rescher, N. (ed.): *The Logic of Decision and Action*. University of Pittsburgh Press, pp. 81–95.

N.E. Fuchs, U. Schwertel, and R. Schwitter. 1998. Attempto Controlled English – Not Just Another Logic Specification Language. In: *Proceedings of LOPSTR'98*, pp. 1–20.

N.E. Fuchs, K. Kaljurand, and T. Kuhn. 2008. Attempto Controlled English for Knowledge Representation. In: *Reasoning Web, Fourth International Summer School 2008*, LNCS 5224, pp. 104–124.

G. Gazdar, C. Mellish. 1989. *Natural Language Processing in Prolog*. An Introduction to Computational Linguistics, Addison-Wesley.

H. Kamp, U. Reyle. 1993. From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory, Kluwer.

M. Kay. 1980. Algorithm Schemata and Data Structures in Syntactic Processing. In: *CSL-80-12*, Xerox Parc, Palo Alto, California.

K. Kipper, A. Korhonen, N. Ryant, and M. Palmer. 2008. A large-scale classification of english verbs. In: Language Resources and Evaluation 42(1), pp. 21–40.

T. Kuhn. 2008. AceWiki: A Natural and Expressive Semantic Wiki. In: *Proceedings of Semantic Web User Interaction at CHI 2008: Exploring HCI Challenges*.

T. Kuhn, R. Schwitter. 2008. Writing Support for Controlled Natural Languages. In: *Proceedings of ALTA 2008*, Tasmania, Australia, pp. 46–54.

T. Parsons. 1994. Events in the Semantics of English. A Study in Subatomic Semantics. MIT Press.

F.C.N. Pereira, S.M. Shieber. 1987. *Prolog and Natural-Language Analysis*. CSLI, Lecture Notes, Number 10.

R. Schwitter. 2002. English as a Formal Specification Language. In: *Proceedings of DEXA 2002*, September 2-6, Aix-en-Provence, France, pp. 228–232.

R. Schwitter, A. Ljungberg, and D. Hood. 2003. ECOLE – A Look-ahead Editor for a Controlled Language, In: *Proceedings of EAMT-CLAW03*, May 15-17, Dublin City University, Ireland, pp. 141–150.

R. Schwitter. 2008. Working for Two: a Bidirectional Grammar for a Controlled Natural Language. In: *LNAI 5360*, pp. 168–179.

J.F. Sowa. 2004. Common Logic Controlled English. *Technical Report*, 24 February 2004. `http://www.jfsowa.com/clce/specs.htm`

G. Sutcliffe, C.B. Suttner. 1998. The TPTP Problem Library: CNF Release v1.2.1. In: Journal of Automated Reasoning, 21(2), pp. 177–203.

H.R. Tennant, K.M. Ross, R.M. Saenz, C.W. Thompson, and J.R. Miller. 1983. Menu-Based Natural Language Understanding. In: *Proceedings of the 21st Meeting of the Association for Computational Linguistics* (ACL), MIT Press, pp. 51–58.

C. Thompson, P. Pazandak, and H. Tennant. 2005. Talk to Your Semantic Web. In: *IEEE Internet Computing*, 9(6), pp. 75–78.

A Sentiment Detection Engine for Internet Stock Message Boards

Christopher C. Chua [†‡] **Maria Milosavljevic** [‡] **James R. Curran** [‡*]

School of Computer Science and Engineering[†]
University of New South Wales
NSW 2052, Australia

Capital Markets CRC Ltd [‡]
55 Harrington Street
NSW 2000, Australia

School of Information Technologies [*]
University of Sydney
NSW 2006, Australia

{cchua,maria}@cmcrc.com james@it.usyd.edu.au

Abstract

Financial investors trade on the basis of information, and in particular, on the likelihood that a piece of information will impact the market. The ability to predict this within milliseconds of the information being released would be useful in applications such as algorithmic trading. We present a solution for classifying investor sentiment on internet stock message boards. Our solution develops upon prior work and examines several approaches for selecting features in a messy and sparse data set. Using a variation of the Bayes classifier with feature selection methods allows us to produce a system with better accuracy, execution performance and precision than using conventional Naïve Bayes and SVM classifiers. Evaluation against author-selected sentiment labels results in an accuracy of 78.72% compared to a human annotation and conventional Naïve Bayes accuracy of 57% and 65.63% respectively.

1 Introduction

In this paper, we present a sentiment prediction engine for classifying investor sentiment, i.e. signals to buy, sell or hold stock positions, based on messages posted on internet stock forums. Our sentiment annotated corpus comes from HotCopper[1], the most popular investment forum for the Australian market, where posts include author self-reported sentiment labels. This unique characteristic of this data set present us with an opportunity to extend research in sentiment classification. Our automated sentiment detection engine implementation uses variations classifiers, particularly the Bernoulli Naïve Bayes and the Complement Naïve Bayes (CNB) models, coupled with feature selection techniques using InfoGain, phrase polarity counts and price alerts. Our methods achieve 78.72% accuracy for CNB and 78.45% for Bernoulli. These figures are higher than the 57% accuracy from human annotators and 65.63% in the baseline. It also outperforms results from Das and Chen (2007) on a different dataset.

[1] http://www.hotcopper.com.au

2 Problem Domain

Our results contribute towards the development of a real-time solution which monitors financial information in order to provide useful advice to support market surveillance analysts' task of explaining alerts surrounding price movements in stocks. For example, when the overall sentiment towards a particular stock is positive, it may well explain the observed increase in its uptake. Many forums do not provide an ability for authors to explicitly report sentiment, thus we hope to eventually apply this model to other forums. The "Buy", "Hold" and "Sell" tags are analogous to positive, neutral and negative sentiments respectively. HotCopper also includes a finer-grained labeling system for "short term" and "long term" sentiments. However such distinctions are beyond our current scope because a finer granularity in the recommendation strength given limited contextual information is often established through an in-depth knowledge of underlying financial fundamentals or information related to a particular stock not reflected within a short message text.

Classifying investor sentiment based on web forum messages is a challenging problem in the text classification domain. The dataset is not only sparse, but varies in the overall quality of its labels and descriptive content. For instance, the sentiment labels are likely to vary in a thread from one post to another, which indicates disagreement. Previous work on sentiment classification is based around relatively well-formed texts (Durant and Smith, 2006; Pang et al., 2002). As demonstrated in Milosavljevic et al. (2007), information extraction techniques such as sentence boundary detection and part-of-speech tagging work relatively well on structured texts but perform less well on messy and sparse data sets such as forum posts and interview transcripts. Hence, we require the use of techniques beyond conventional approaches. Furthermore, the constraints of a real-time classification system presents additional challenges.

3 Background

As the literature directly related to this domain is limited, we draw from related areas of sentiment classification research where a research efforts have been concentrated around sentiment or opinion analysis for political blogs (Durant and Smith, 2006) and product reviews (Yi et al., 2003). The methods developed in those prior work are relevant to our application.

Sentiment analysis on web forums specifically within the financial domain has also been investigated by Das and Chen (2007). Their focus, like ours, is on capturing the emotive aspect of the text rather than the factual content. In their research, the Naïve Bayes (NB) classifier is found to yield the best results, and a voting mechanism is used in conjunction with additional classifiers such as SVM to improve accuracy. However, the classification accuracy achieved at 62% using a simple majority vote of multiple classifiers with a small sample and the low inter-annotator agreement demonstrate the difficulty in classifying such datasets. Antweiler and Frank (2004)'s research findings found that online forum discussions between investors are not equivalent to market noise, and instead contain financially-relevant informational content. As a result, effective sentiment detection can predict market volume and volatility across stocks, thus highlighting the need for placing such web discussions under the investigative eyes of surveillance analysts. Both Das and Chen (2007) and Antweiler and Frank (2004) use data from Yahoo Finance and Raging Bull based in the US, covering only a subset of stocks, with classification performed per stock rather than in aggregate.

The prior literature demonstrates that the sentiment analysis task can be performed using a variety of classification methods, chief among them the NB model (Das and Chen, 2007; Antweiler and Frank, 2004). Similar to Das and Chen (2007) and Antweiler and Frank (2004), we find that a typical SVM classifier performs no better than the alternatives we attempted, while suffering from a higher degree of complexity affecting execution performance. Moreover, prior solutions presented do not offer a comprehensive sentiment analyser to predict sentiment off financial forums in real-time for market surveillance or technical trading. We extend the concepts presented in prior research by incorporating additional contextual information in our training tasks, developing more advanced feature selection as well as adopting variations of the models used in related research.

Statistic	Buy	Sell	Hold
Total	6379	469	1459
Monthly Average	1063.17	78.17	243.17
Monthly Std Dev	283.65	28.22	50.95

Table 1: HotCopper Post Statistics

4 Data

In our analysis, we use the first six months of 2004 HotCopper ASX stock-based discussions. There are 8,307 labeled posts across 469 stocks, with an average of 28 words per post and a total of 23,670 distinct words in the dataset. Each message is organised by thread, with a single thread consisting of multiple posts on the same topic for a stock. We consider both long term and short term variations of a sentiment to be equivalent. "Buy" recommendations outnumber "Sell" and "Hold" 13.6 and 4.4 times respectively. Within first 18 months of our analysis, the average monthly posts increased from over 1,400 to a peak of over 3,700 posts by August 2005, indicating growing forum participation. Discussions on HotCopper mainly surrounds speculative stocks, particularly those in minerals exploration and energy. In

fact, some of the biggest stocks by market capitalisation on the ASX such as the Commonwealth Bank (CBA) and Woolworths (WOW) generate little to no active discussions on the forums, highlighting the focus on small and speculative stocks.

4.1 Data Preprocessing

We perform a series of preprocessing steps for each post to obtain a canonical representation, firstly by removing stop words from the training set in the NLTK stop list (Bird et al., 2009). Words and alphanumerics of non-informative value, e.g. "aaaaaa" or "aaaaah", are filtered out, the remaining stemmed using the Porter algorithm (Porter, 2009) with spell-correction applied using the first suggestion from the PyEnchant package (Kelly, 2009).

We observed many ambiguous instances which introduce noise to the training model. In order to control for this, a thread volatility measure is introduced for the message where we assign an index value representing the level of disagreement between subsequent replies in the thread. The thread volatility is measured as the average sum of the differences between the discretised values of the sentiment classes. We assign buy and sell to have the furthest distance, thus the discretised set S contains $\{$buy=1,hold=2,sell=3$\}$. Threads which transitions from buy to sell result in a higher volatility figure than threads which transitions from buy to hold. This allows for the posts within a thread with lower volatility to emerge as a superior sample. The thread volatility measure for a discretised sentiment s_i in thread t with N_t posts, is defined as follow:

$$\sigma_t = \frac{1}{N_t} \sum_{i=1}^{N_t - 1} |s_{i+1} - s_i|$$

We select threads with low volatility (< 0.5) for our training base in order to reduce the level of disagreement in the training set. This filtering step reduces our effective sample size to 7,584 and enhances the quality of the training sample.

5 Classification

Our first experiment consisted of a baseline NB classifier (McCallum, 1998). The NB classifier follows Bayesian probability theory in selecting the maximum likelihood of an outcome given its prior probabilities. We are interested in the most probable class (MAP), given a message instance d with n features f and set of sentiment classes S:

$$MAP = \arg\max_{s \in S} P(s) \prod_{i=1}^{n} P(f_i|s)$$

A simplifying assumption is to treat the presence of individual features in the message d containing n words as positionally-independent of other words in the document. Although weakly-formed, this is found to perform well due to its zero-one loss property (Domingos and Pazzani, 1997). Laplace's add-one smoothing method is used to account for zero probabilities.

Following this, we tested an adapted version of the NB classifier to improve our classification accuracy, by incorporating the Term Frequency Inverse Document Frequency (TF-IDF) transformation (Rennie et al., 2003), which allows us to weigh terms that provide a greater distinction to a particular post more heavily than ones which appear with regular frequency across all posts and are poor features to rely on for classification.

$$TF - IDF_{f_i} = \ln\left(\sum f_i + 1\right) \ln\left(\frac{\sum_j d_j}{\sum_j d_{j,s \in S}}\right)$$

Another issue that we have to contend with is the uneven class distribution in the dataset, which is a common issue in text categorisation. Undersampling or oversampling methods results in an inaccurate distribution of underlying data, hence to overcome this limitation, we apply the approach used by Rennie et al. (2003) to tackle this skewness. The CNB classifier improves upon the weakness of the typical NB classifier by estimating parameters from data in all sentiment classes except the one which we are evaluating for. For a given message j with n features f, the CNB classifies documents according to the following rule:

$$l(f) = \arg\min_{s \in S} \sum_{i=1}^{n} f_i w_{s_i}$$

f_i is the count of feature i in the post and w_{s_i} is the complement weight parameter which is the TF-IDF transformed complement of the likelihood estimates (see Rennie et al. (2003)).

Finally, we also tested the classifier performance with the Bernoulli model of Naïve Bayes (McCallum, 1998), which replaces feature frequency counts with Boolean values. The use of the CNB classifier

Classifier	NB Baseline	CNB	CNB IG	NB Binarised	NB Binarised IG
# of Features	7,200	7,200	1205 (Rank 50)	7,200	1205 (Rank 50)
Accuracy	65.63%	74.41%	**78.72%**	75.75%	78.45%
Precision	68.50%	74.80%	**76.70%**	70.30%	73.40%
Recall	65.60%	74.40%	78.70%	75.80%	**78.50%**
F-score	66.90%	74.50%	**77.50%**	72.00%	72.00%

Table 2: Results Summary

and Bernoulli variant yields a statistically significant improvement in the classification accuracy, which is consistent with the findings of Pang et al. (2002) in the sentiment analysis domain.

6 Feature Selection

The features are first ranked by order of frequency. An optimal set of features is selected by testing feature increments up to a maximum of 10,000 features; approximately 40% of the base. We then tested the information gain (InfoGain) algorithm (Yang and Pedersen, 1997), which is useful in filtering out the vast number of features to a manageable subset. Among the additional features we incorporate is the count of positive and negative bigrams and trigrams (including negations) of the form "ADJ financial_term" where financial terms are common phrases encountered within the sample such as "EPS", "dividends" and "profit" representing domain-specific knowledge. Another domain-specific feature we incorporate is the count of stock price alerts in the 3 days preceding the start of a thread. A price risefall alert is triggered when the stock price risesdrops beyond 4 standard deviations from its historical price change levels.

7 Results and Evaluation

In any machine learning task, it is crucial to verify our results against human agreement levels. We took a random sample of 100 opening posts (to avoid out of context replies) and published an annotation task using Amazon's Mechanical Turk (MTurk) (Amazon, 2009) to obtain classifications from three paid annotators who passed a test. The disappointingly low annotator accuracy of 57% and Kappa agreement of 50% demonstrates the challenging nature of this task, even for humans.

We perform each experiment using 10-fold cross-validation and compare the performance based on accuracy in conjunction with F-scores. Table 2 summarises our main findings in terms of sentiment classification quality. At 7,200 features, the best performance is seen in the CNB and Bernoulli classifiers. In both schemes, InfoGain attribute selection improved F-scores by 10.60% and 5.10% respectively with 1,205 features compared to the baseline. The overall accuracy of both classifiers, at 78.72% and 78.45% are significantly above those attained in the baseline.

Our results reveal two classification strategies in our implementation, i.e. using either the CNB or the Bernoulli NB model. We also find that feature selection techniques and filtering noisy instances with the volatility measure, increase overall performance to a level higher than that of the baseline. Positive and negative phrase counts do not yield significant improvements in performance, which could be explained by a change in sentiment tone as evidenced in Pang et al. (2002). For example, a post may be labeled "Sell" but contain positive messages unrelated to the subject. This may be improved by using entity recognition to disambiguate context. Further extension that we hope to incorporate into the classification model is the addition of financial information reported in the media to help augment information not reflected in the message board post.

8 Conclusion

We introduce a sentiment prediction engine that allows for the real-time classification of sentiment on internet stock message boards. Through the application of alternative models and additional feature selection schemes, we are able to achieve classification F-score of up to 77.50%. We believe that more advanced natural language processing techniques, particularly deeper contextual analysis using external sources of financial data as well as improving the handling of imbalanced classes, will provide fruitful grounds for future research.

References

Amazon. 2009. Amazon Mechanical Turk. http://aws.amazon.com/mturk.

W. Antweiler and M.Z. Frank. 2004. Is All that Talk just Noise? The Information Content of Internet Stock Message Boards. *Journal of Finance*, pages 1259–1294.

S. Bird, E. Loper, and E. Klein. 2009. Natural Language Toolkit. http://www.nltk.org.

S.R. Das and M.Y. Chen. 2007. Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web. *Journal of Management Science*, 53(9):1375–1388.

P. Domingos and M. Pazzani. 1997. Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. *Machine Learning*, 29(2–3):103–130.

K.T. Durant and M.D. Smith. 2006. Mining Sentiment Classification from Political Web Logs. In *Proceedings of Workshop on Web Mining and Web Usage Analysis of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (WebKDD-2006), Philadelphia, PA*.

R. Kelly. 2009. PyEnchant Spellchecker. http://www.rfk.id.au/software/pyenchant/.

A. McCallum. 1998. A Comparison of Event Models for Naïve Bayes Text Classification. In *AAAI Workshop on Learning for Text Categorization*, pages 41–48.

M. Milosavljevic, C. Grover, and L. Corti. 2007. Smart Qualitative Data (SQUAD): Information Extraction in a Large Document Archive. In *Proceedings of the 8th RIAO Conference*.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs Up?: Sentiment Classification using Machine Learning Techniques. In *Proceedings of the ACL conference on Empirical Methods in Natural Language Processing*, volume 10, pages 79–86. Association for Computational Linguistics.

M. Porter. 2009. The Porter Stemming Algorithm. http://tartarus.org/~martin/PorterStemmer/.

J.D. Rennie, L. Shih, J. Teevan, and D. Karger. 2003. Tackling the Poor Assumptions of Naïve Bayes Text Classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 616–623.

Y. Yang and J.O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420.

J. Yi, T. Nasukawa, R. Bunescu, W. Niblack, I.B.M.A.R. Center, and CA San Jose. 2003. Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. In *Third IEEE International Conference on Data Mining*, pages 427–434.

An Unsupervised Approach to Domain-Specific Term Extraction

Su Nam Kim[♠], **Timothy Baldwin**[♡]
[♠] [♡] CSSE
[♡] NICTA VRL
University of Melbourne
VIC 3010 Australia
sunamkim@gmail.com, tb@ldwin.net

Min-Yen Kan[♣]
[♣] School of Computing
National University of Singapore
kanmy@comp.nus.edu.sg

Abstract

Domain-specific terms provide vital semantic information for many natural language processing (NLP) tasks and applications, but remain a largely untapped resource in the field. In this paper, we propose an unsupervised method to extract domain-specific terms from the Reuters document collection using term frequency and inverse document frequency.

1 Introduction

Automatic domain-specific term extraction is a categorization/classification task where terms are categorized into a set of predefined domains. It has been employed in tasks such as keyphrase extraction (Frank et al., 1999; Witten et al., 1999), word sense disambiguation (Magnini et al., 2002), and query expansion and cross-lingual text categorization (Rigutini et al., 2005). Even though the approach shows promise, relatively little research has been carried out to study its effects in detail (Drouin, 2004; Milne et al., 2006; Rigutini et al., 2006; Kida et al., 2007; Park et al., 2008). Most of the research to date on domain-specific term extraction has employed supervised machine learning, within the fields of term categorization and text mining. However, to date, the only research to approach the task in an unsupervised manner is that of Park et al. (2008). Unsupervised methods have the obvious advantage that they circumvent the need for laborious manual classification of training instances, and are thus readily applicable to arbitrary sets of domains, tasks and languages.

In this paper, we present a novel unsupervised method for automatically extracting domain-specific terms, targeted specifically at building domain-specific lexicons for natural language processing (NLP) purposes. One of the main properties utilized in this work is *domain specificity*. Our notion of *domain specificity* is based on statistical analysis of word usage, and adopts the simple notions of *term frequency* (*TF*) and *inverse document frequency* (*IDF*) over domains to capture their domain specificity.

2 Unsupervised Domain-Specific Term Extraction

In this section, we elaborate on our proposed method, as well as the benchmark method of Park et al. (2008).

2.1 Proposed Method (D1)

Our proposed unsupervised method is based on *TF-IDF*. The basic underlying idea is that domain-specific terms occur in a particular domain with markedly higher frequency than they do in other domains, similar to term frequency patterns captured by *TF-IDF*.

Hence, we compute *TF-IDF* from TF_{ij}, the term frequency of term i from documents in *domain* j, and IDF_i, the inverse domain frequency. The calculation of TF_{ij} is via:

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \quad (1)$$

where n_{ij} is the number of occurrences of term i in the documents associated with domain j. IDF_i is calculated via:

$$IDF_i = log(\frac{|D|}{|\{d : t_i \in d\}|}) \quad (2)$$

where t_i is the term, and D is the set of all domains.

The final *TF-IDF*$_{ij}$ value of a given term is the simple product of TF_{ij} and IDF_i.

Once the task of scoring terms has been completed, we select those terms which have higher

values than a given threshold. We select the threshold heuristically based on the score distribution, specifically choosing the point at which there is a significant drop in *TF-IDF* scores. That is, when the number of domain-specific terms gained at the current similarity is no more than 20% of the previously-accumulated domain-specific terms, we use that similarity as our threshold.

2.2 Benchmark Method (D2)

We compare our proposed method with the only unsupervised domain-specific term extraction method, i.e. the method of Park et al. (2008). Park *et al.* directly compare term frequencies in documents for a given domain d with term frequencies in the general document collection, based on:

$$domain\_specificity(w) = \frac{\frac{c_d(w)}{N_d}}{\frac{c_g(w)}{N_g}} \quad (3)$$

where $c_d(w)$ and $c_g(w)$ denote the number of occurrences of term w in the domain text and general document collection, respectively. N_d and N_g are the numbers of terms in the domain-specific corpus and in the general corpus, respectively. If term w does not occur in the general corpus, then $c_g(w)$ is set to 1; otherwise it is set to the highest count in the general corpus. In the original work, a one million term corpus from mostly news articles was used as the general corpus. The final score is computed as:

$$Final\ Score = \frac{F + M}{N} \times 100 \quad (4)$$

where N is the number of keywords in the reference data, F is the number of falsely-recognized domain-specific terms (false positives), and M is the number of missed domain-specific terms (false negatives). We avoid computing the final score as shown in (4) since we do not have reference data. Instead, we set a threshold by looking for a significant drop in the score (i.e. the score when the number of newly-extracted terms is less than 20% of the previously-learned terms), as in our approach (D1).

2.3 Collecting Domain-Specific Words

To collect domain-specific words, we used *the modified Lewis split* of the *Reuters document collection*,[1] a dataset which has been extensively used for text categorization, since it contains document-level topics (i.e. domains). In detail, *the modified Lewis split* version of the collection is made up of 90 topics and 3,019 and 7,771 test and training documents, respectively. We extract domain-specific terms from the training documents, and use the 3,019 test articles for text categorization and keyphrase extraction evaluation in Section 3.

Domain	D1	D2	Domain	D1	D2
platinum	132	62	oat	115	49
lumber	77	165	lead	71	105
orange	69	160	hog	61	106
pet-chem	55	246	strategic-metal	50	136
income	49	64	fuel	42	80
alum	37	316	rapeseed	35	13
heat	35	58	tin	33	222
silver	29	99	copper	22	236
wpi	20	87	soy-oil	17	18
zinc	14	50	rubber	13	369
gas	13	122	soy-meal	12	23
meal-feed	12	85			

Table 1: Number of extracted domain-specific terms

After collecting words with the proposed (**D1**) and benchmark (**D2**) methods, we compared them in the form of the ratio of domain-specific terms to the number of domains. Among all the domains present in the corpus, we selected 23 domains which had at least 5 articles in both the test and training data splits, both to manually verify the performance of the two methods, and to utilize the collected domain-specific terms in applications. The total number of terms collected from the 386 selected articles were 1,013 and 2,865, respectively. Table 1 shows the number of domain-specific terms extracted by D1 and the method of D2 over the selected 23 domains. D2 extracts nearly three times more domain-specific terms than D1, but the distribution of terms across domains is relatively well proportioned with D1. This preliminary observation suggests that D1 is more reliable than the benchmark system.

2.4 Human Verification

We manually verified how well our proposed method extracts domain-specific terms. Unlike the method of (Drouin, 2004), where experts

[1] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

scored extracted terms for subtle differences in domain specificity, we opted for a simple annotation process involving non-expert annotators. We asked three human annotators to answer "yes" or "no" when given a term and its domain, as predicted by the two methods. Note that before annotating the actual data set, we trained the human annotators in the context of a pilot annotation test. In our human verification process, we attained an accuracy of 40.59% and 36.59% for D1 and D2, respectively, with initial inter-annotator agreement of 69.61% and 73.04%, respectively. Thus, we cautiously conclude that our proposed method performs better than Park et al. (2008). Note that as the work in Park et al. (2008) was originally developed to extract domain-specific terms for use in correcting domain term errors, the authors did not discuss the performance of domain-specific term extraction in isolation.

We made a few observations during the manual verification process. Despite the strict annotation guidelines (which were further adjusted after the pilot test), the agreement actually dropped between the pilot and the final annotation (especially with one of the annotators, namely A_3). We asked the individual annotators about the ease of the verification procedure and the notion of domain specificity, from their individual perspective. It turned out that although the choice of domain specificity was guided by statistical usage, word senses were involved in the decision process to some degree. Additionally, the annotators commented on the subjectivity of statistical markedness of the terms. The average correlations among two annotators are .738 and .673 for D1 and D2, respectively.

3 Applying Domain-Specific Terms

In this section, we evaluate the utility of the collected domain-specific terms via two tasks: text categorization and keyphrase extraction. Our motivation in selecting these tasks is that domain-specific terms should offer a better representation of document topic than general terms.

3.1 Text Categorization

Automatic text categorization is the task of classifying documents into a set of predefined categories.

Type	F1		F3	
	TF	*TF-IDF*	*TF*	*TF-IDF*
Baseline	.473	.660	.477	.677
Domain	.536	.587	–	–
Combined	.583	**.681**	.579	**.681**

Table 2: Performance of text categorization

To build a text categorization model, we first preprocess the documents, perform part-of-speech (POS) tagging using a probabilistic POS tagger,[2] and lemmatization using `morpha` (Minnen et al., 2001). We then build an SVM-based classifier (Joachims, 1998).[3] We use *TF-IDF* for feature weighting, and all unigram terms. Note that when domain-specific terms are multiword noun phrases (NP), we break them down into unigrams based on the findings of Hulth and Megyesi (2006). As baselines, we built systems using unigram terms which occur above a threshold frequency (i.e. frequency $\geq$ 1, 2 and 3 or F1, F2, F3 in Table 2) after removing stop words. Table 2 shows the micro-averaged F-scores of the text categorization task. Note that since using F2 results in the lowest performance, we report only results over thresholds F1 and F3.

Table 2 shows that domain-specific terms alone do not perform well, since only a relatively small volume of domain-specific indexing terms are extracted, compared to the number of unigram terms. However, when combined with a unigram model, they aid unigram models to improve the overall performance. Despite only showing a small improvement, given the relatively small number of domain-specific terms extracted by our method, we confirm that domain-specific terms are useful for categorizing (monolingual) texts, just as domain specificity has been shown to help in cross-lingual text categorization (Rigutini et al., 2005).

3.2 Automatic Keyphrase Extraction

Keyphrases are simplex nouns or NPs that represent the key ideas of the document. They can serve as a representative summary of the document and also as high-quality index terms. In the past, various attempts have been made

[2] `Lingua::EN::Tagger`

[3] `http://svmlight.joachims.org/svm_multiclass.html`

Type	L	*Boolean*	*TF*	*TF-IDF*
KEA	NB	.200	–	–
	ME	.249	–	–
KEA +	NB	.204	.200	.197
Domain	ME	.260	.261	**.267**

Table 3: Performance of keyphrase extraction

to boost automatic keyphrase extraction performance, based primarily on statistics (Frank et al., 1999; Witten et al., 1999) and a rich set of heuristic features (Nguyen and Kan, 2007).

To collect the gold-standard keyphrases, we hired two human annotators to manually assign keyphrases to 210 test articles in the same 23 selected domains. In summary, we collected a total of 1,339 keyphrases containing 911 simplex keyphrases and 428 NPs. We checked the keyphrases found after applying the candidate selection method employed from Nguyen and Kan (2007). The final number of keyphrases found in our data was only 750 (56.01% of all the documents), among which 158 (21.07%) were NPs.

To build a keyphrase extractor, we first preprocessed them with a POS tagger and lemmatizer, and applied the candidate selection method in Nguyen and Kan (2007) to extract candidates. Then, we adopted two features from KEA (Frank et al., 1999; Witten et al., 1999), as well as the domain-specific terms collected by our method. KEA uses two commonly used features: *TF-IDF* for document cohesion, and *distance* to model the locality of keyphrases. Finally, we used the features to build a maxent classifier[4] and a Naïve Bayes (NB) model. To represent the domain specificity of the keyphrase candidates, we simply presented the 23 domains as three separate sets of features with differing values (*Boolean*, *TF* and *TF-IDF*), when a given keyphrase candidate is indeed a domain-specific term. Finally, with KEA as a baseline, we compared the systems over the top-7 candidates using the current standard evaluation method (i.e. *exact matching scheme*). Table 3 shows the micro-averaged F-scores.

In the results, we first notice that our test system outperformed KEA with ME, but that our test system using *Boolean* produced better performance than KEA only with NB. The maximum improvement in F-score is about 1.8%, in the best configuration where *TF-IDF* weighting is used in conjunction with an ME learner. This is particularly notable because: (a) the average performance of current keyphrase extraction systems is a little more than 3 matching keyphrases over the top 15 candidates, but we produce only 7 candidates; and (b) the candidate selection method we employed (Nguyen and Kan, 2007) found only 56.01% of keyphrases as candidates. Finally, we note with cautious optimism that domain-specific terms can help in keyphrase extraction, since keyphrases are similar in the same or similar domains.

[4] http://maxent.sourceforge.net/index.html

4 Conclusions

In this work, we have presented an unsupervised method which automatically extracts domain-specific terms based on term and document statistics, using a simple adaptation of *TF-IDF*. We compared our method with the benchmark method of Park et al. (2008) using human judgements. Although our method did not extract a large number of domain-specific terms, the quality of terms is high and well distributed over all domains. In addition, we have confirmed the utility of domain-specific terms in both text categorization and keyphrase extraction tasks. We empirically verified that domain-specific terms are indeed useful in keyphrase extraction, and to a lesser degree, text categorization. Although we could not conclusively prove the higher utility of these terms, there is a strong indicator that they are useful and deserve further analysis. Additionally, given the small number of domain-specific terms we extracted and used, we conclude that they are useful for text categorization.

Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

K.V. Chandrinos and I. Androutsopoulos and G. Paliouras and C.D. Spyropoulos. Automatic Web

rating: Filtering obscence content on the Web. In *Proceedings of ECRATDL*. 2000, pp. 403–406.

P. Drouin. Detection of Domain Specific Terminology Using Corpora Comparison. In *Proceedings of LREC*. 2004, pp. 79–82.

G. Escudero and L. Marquez and G. Rigau. Boosting applied to word sense disambiguation. In *Proceedings of 11th ECML*. 2000, pp. 129–141.

E. Frank and G.W. Paynter and I. Witten and C. Gutwin and C.G. Nevill-Manning. Domain Specific Keyphrase Extraction. In *Proceedings of 16th IJCAI*. 1999, pp. 668–673.

A. Hulth and B. Megayesi. A Study on Automatically Extracted Keywords in Text Categorization. In *Proceedings of COLING/ACL*. 2006, pp. 537–544.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of ECML*. 1998, pp. 137–142.

M. Kida and M. Tonoike and T. Utsuro and S. Sato. Domain Classification of Technical Terms Using the Web. *Systems and Computers*. 2007, 38(14), pp. 2470–2482.

P. Lafon. Sur la variabilite de la frequence des formes dans un corpus. In *MOTS*. 1980, pp. 128–165.

B. Magnini and C. Strapparava and G. Pezzulo and A. Gliozzo. The role of domain information in word sense disambiguation. *NLE*. 2002, 8(4), pp. 359–373.

D. Milne and O. Medelyan and I.H. Witten. Mining Domain-Specific Thesauri from Wikipedia: A case study. In *Proceedings of the International Conference on Web Intelligence*. 2006.

G. Minnen and J. Carroll and D. Pearce. Applied morphological processing of English. *NLE*. 2001, 7(3), pp.207–223.

T.D. Nguyen and M.Y. Kan. Key phrase Extraction in Scientific Publications. In *Proceeding of ICADL*. 2007, pp. 317-326.

Y. Park and S. Patwardhan and K. Visweswariah and S.C. Gates. An Empirical Analysis of Word Error Rate and Keyword Error Rate. In *Proceedings of ICSLP*. 2008.

L. Rigutini and B. Liu and Marco Magnini. An EM based training algorithm for cross-language text categorization. In *Proceedings of WI*. 2005, pp. 529-535.

L. Rigutini and E. Di Iorio and M. Ernandes and M. Maggini. Automatic term categorization by extracting knowledge from the Web. In *Proceedings of 17th ECAI*. 2006.

G. Salton and A. Wong and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*. 1975, 18(11), pp. 61–620.

I. Witten and G. Paynter and E. Frank and C. Gutwin and G. Nevill-Manning. KEA:Practical Automatic Key phrase Extraction. In *Proceedings of ACM DL*. 1999, pp. 254–256.

A Cascade Approach to Extracting Medication Events

Jon Patrick
School of IT
The University of Sydney
Sydney, NSW 2006, Australia
jonpat@it.usyd.edu.au

Min Li
School of IT
The University of Sydney
Sydney, NSW 2006, Australia
mili9528@uni.sydney.edu.au

Abstract

Information Extraction, from the electronic clinical record is a comparatively new topic for computational linguists. In order to utilize the records to improve the efficiency and quality of health care, the knowledge content should be automatically encoded; however this poses a number of challenges for Natural Language Processing (NLP). In this paper, we present a cascade approach to discover the medication-related information (MEDICATION, DOSAGE, MODE, FREQUENCY, DURATION, REASON, and CONTEXT) from narrative patient records. The prototype of this system was used to participate the i2b2 2009 medication extraction challenge. The results show better than 90% accuracy on 5 out of 7 entities used in the study.

1 Introduction

Electronic records are widely used in the health care domain since we believe they can provide more advantages than the traditional paper record (Sujansky, 1998). However, the value of electronic clinical records depends significantly on our ability to discover and utilize the specific content found in them. Once this content can be detected, the potential benefits for individual clinicians and healthcare organizations are considerable.

In this study we focus on discharge summaries, which have their own challenges. This kind of clinical record includes several sections. The average word count of these reports is around 1500 words per record. This paper presents a method to extract all the medication related information, and connect the relative entities together to build medication entries using a cascaded approach based on two machine learners.

2 Related Work

In this paper, we focus on both NER and RC tasks to extract the medications and their related information (DOSAGE, MODE, FREQUENCY, DURATION, REASON, and CONTEXT) from free-text clinical records. At this time, it's difficult to compare our system with other systems which participated the i2b2 2009 medication extraction challenge, since these publications are unavailable now. Consequently, we can only compare our system with some similar studies in the literature. In the previous work, only three published studies address this issue (see the performance comparison in the final section) and these studies do not have a comprehensive and precise definition of medication information. The closest research for medication event extraction relies on parsing rules written as a set of regular expressions and a user-configurable drug lexicon. It includes the event for DRUG, DOSAGE, ROUTE, FREQUENCY, CONTEXT and NECESSITY (Gold et al. 2008). The basic work flow for their system starts by discovering drug names based on a drug dictionary, and the rest of the process uses the MERKI parser.

The CLARIT NLP system (Evans et al. 1996) can extract DRUG-DOSAGE information from clinical narratives. This system is based on the rule-based method and five main steps are included, such as tokenization, stemming, syntactic category assignment, semantic category assignment and pattern matching.

Another system focuses on the drug extraction only and is based on a drug lexicon (Sirohi and Peissig 2005). This study demonstrates that high precision and recall for medication extraction from clinical records can be obtained by using a carefully selected drug lexicon.

Comparing these three medication extraction systems, a different approach is adopted in our work. Our medication event system is based on the combination of a machine learner approach and rule based approach. Two machine learners were used, namely the conditional random field (CRF) and support vector machine (SVM). Moreover, a broader definition for a medication event is considered, especially the REASON for the medication which hasn't been studied in previous research. Furthermore, the medication information in our training and test set is much larger than prior studies.

3 Methodology

There are four main steps in our methodology:

1. Definition of the information to be extracted.
2. Preparing data for training and testing.
3. Using natural language processing technologies to build a medication event extraction system.
4. Passing the test data to the system and evaluation of the final result.

3.1 Extraction Definition

Our goal is to provide accurate, comprehensive information about the medications a patient has been administered based on the evidence appearing in the textual records. For each medication entry, the following information needs to be extracted: Medication, dosage, mode, frequency, duration, reason, and context.

Multiple medication entries should be generated if the MEDICATION has the changes for DOSAGE or multiple DOSAGEs, MODEs, FREQUENCYs, DURATIONs and REASONs.

3.2 Data Preparation

One hundred and sixty clinical records were prepared for training (130 records) and testing (30 records). One physician and one researcher created the gold standard annotations by sequential annotation: the physician annotated the records first and his results were given to the researcher to revise. The annotation process took approximately 1.5 hours per record due to the length of clinical records.

4 Medication Event Extraction System Architecture

The basic strategy for the medication event extraction system is to: ① use CRF to identify the entities, ② build pairs for each medication relationship (only consider DRUG and its related entity, since the whole related entities, such as DOSAGE, FREQUENTY, etc., could be further connected based on the DRUG), ③ classify the binary relationships by SVM, ④ generate medication entries based on the results from the CRF and SVM. Figure 1 demonstrates the detailed system architecture, which includes the following processing stages:

I. Sentence Spitting

Split the clinical records into individual sentences.

II. Tokenization

Each sentence is split into tokens their position and extent in the text.

III. CRF Feature Builder

Seven feature sets were prepared in this stage, to be used in the CRF training. They are DRUG, DOSAGE, MODE, FREQUENCY, DURATION, REASON, and morphology feature sets.

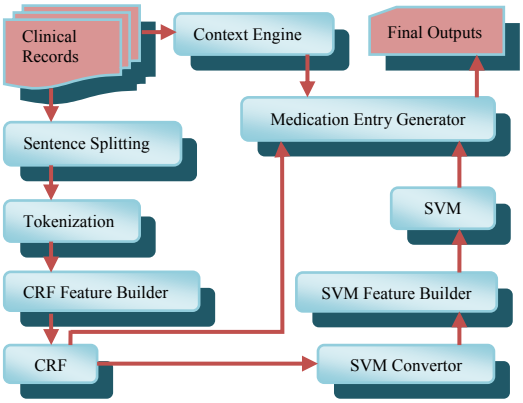

Figure 1. NER and RC System Architecture.

IV. CRF Model Building and Classification

The CRF feature builder generated the features for the CRF machine learner. The context window for the CRF was set to be five words.

V. CRF Model Building and Classification

The CRF results were converted into SVM input features by the SVM Convertor. There are two kinds of SVM input generated here:

1. Unigram Sentences

Each pair of medication elements at the unigram sentence level is used to build an SVM training record.

2. Sentence Pairs

Sometimes MEDICATION and its REASON could be across two sentences. Like the mechanism to generate the unigram sentence input, medication pairs are also built at the sentence pair level.

VI. SVM Feature Builder

Six Features are generated based on the output from the SVM Convertor to classify the relationships:

1. Three words before and after the first entity.
2. Three words before and after the second entity.
3. Words between the two entities.
4. Words inside of each entity.
5. The types of the two entities determined by the CRF classifier.
6. The entities types between the two entities.

VII. SVM Model Building and Classification

The features which were generated in the previous step were passed to the SVM to build the model and classify the relationships between medication pairs for the test set.

VIII. CONTEXT Identification

The CONTEXT engine identifies the medication entry under the special section headings, such as "MEDICATIONS ON ADMISSION:", "DISCHARGE MEDICATIONS:" etc., or in the narrative part of the clinical record. The performance is discussed in the next section.

IX. Medication Entry Generation

The medication entry generator is the final step in this system which is responsible for assembling all

the components into the final medication event entries based on having established their relationships. The results from the previous steps are used here, namely CRF, SVM and CONTEXT Engine. Two stages are involved in this step:

(a) Using the SVM results to identify the medication entries. The CONTEXT value (list/narrative) comes from the CONTEXT Engine. The algorithm which is used to build medication entries is based on the position rule of each entity and the total number of each entity type. It can be divided into several cases.

(b) If the medication in the clinical notes doesn't have any relationships with other entity types, it will be missing from the SVM result. Consequently, this medication should be withdrawn from the CRF results and an individual medication entry generated for it. The value for the CONTEXT (list/narrative) also comes from the CONTEXT Engine, as in the previous step.

5 Results and Discussion

In this section, the experiment results for NER, RC, CONTEXT engine and the final output for the test set is presented and discussed.

5.1 NER(CRF) Experiment

The main purpose of this experiment is to extract the MEDICATION, DOSAGE, MODE, FREQUENCY, DURATION and REASON from the clinical records. Table 1 demonstrates the performances for exact match by using the 7 feature sets. The number in the bracket is the baseline, which use the bag of words as the only feature set. The baseline shows the extraction for REASON and DURATION are the most difficult entities to recognise (their average F-score is about 50%, while the MODE, DOSAGE and FREQUENCY perform best with an average F-score greater than 92%).

Entity Type	Training	Test	Recall (Baseline)	Precision (Baseline)	F-Score (Baseline)
Overall	**17337**	**5296**	**88.82% (80.25%)**	**92.89% (93.49%)**	**90.81% (86.36%)**
MEDICATION	6576	1940	91.44% (76.34%)	91.35% (91.87%)	91.40% (83.39%)
DOSAGE	3352	1076	93.49% (88.66%)	96.36% (95.69%)	94.91% (92.04%)
MODE	2537	796	94.60% (91.21%)	95.92% (96.93%)	95.26% (93.98%)
FREQUENCY	3180	1020	93.24% (90.26%)	96.26% (95.74%)	94.72% (92.94%)
DURATION	366	104	51.92% (41.35%)	80.60% (79.63%)	63.16% (54.43%)
REASON	1326	360	46.11% (34.72%)	69.75% (72.67%)	55.52% (46.99%)

Table 1. Best scores and baseline scores from CRF of NER

It is worth pointing out many other features were experimented with during the system implementation, such as the medical category for each word, whether the word is capitalized, in lower case or upper case, etc. However, the best performance is obtained from the 7 feature set. The feature selection process is that:

In the first place, all features were gathered together to train the model and predict the results. Sequentially, the performance of this experiment was recorded. Next, we did a set of experiments to remove every feature from the whole features one by one, and then train the related model. After that, predict the results and record the performance. Finally, these performances were compared with the performance in the first step to see whether the removed feature decreased in the F-score. If it did, this feature would be useful. Else, it was useless.

The performances for the REASON and DURATION are still the lowest, but the F-scores are approximately 10% higher than the baseline. This is because:

1. The frequencies for the REASON and DURATION are much smaller than the other four entity types.
2. For the DURATION entities, the rule based regular expression can match other non-medication terms. Also, there are some DURATION terms that can't be discovered by our rules.
3. REASON extraction depends highly on the Finding category in SNOMED CT and the performance of TTSCT (Patrick et al. 2007). However, the Finding category cannot be well-matched to the REASON entities in the clinical notes, due to the many varied ways REASON can be represented which may not exist in the SNOMED CT, and as well the REASONs that are ambiguously expressed. Another limitation is the performance of TTSCT. Consequently, these issues lead to low performance on REASON, and the F-score of DURATION (63.16%) is higher than the REASON (55.52%) even though the frequency of DURATION is smaller than REASON (104 and 360 respectively).

Compared to the baseline, the F-scores for the MODE, DOSAGE and FREQUENCY were only improved by about 2%. The first reason is that the performance of the baseline is already very high (around 90%). Secondly, the regular expressions and gazetteers cannot capture all the different ways to present these three entity types. Approximately 8% improvement in the MEDICATION extraction is obtained in the system, since the medication lexica were used in the system. The errors come from:

1. Misspelling of drug names, such as "nitroglycerin"
2. Drug names used in other contexts, such as the "coumadin" in the "Coumadin Clinic" phrase.
3. The drug allergies detector cannot cover all situations.

Overall, the system scored of 90.81% on the NER task.

5.2 Relationship Classification Experiment

The support vector machine is used to classify the relationships between the medication pairs (see section 2). The feature sets used are discussed in the previous section. Meanwhile, the feature selection mechanism is same as the NER feature selection, which was introduced in the previous sub-section. For comparison, the baseline only uses three of the whole feature sets, namely, No.1, 2 and 4 in the SVM feature sets. Two experiments were conducted (the unigram sentence level and sentence pair level) for the baseline and subsequent solutions.

Relation Type	Total Number	Recall (Baseline)	Precision (Baseline)	F-Score (Baseline)
HAS RELATIONSHIP (unigram)	**3373**	**98.89% (82.69%)**	**97.90% (61.26%)**	**98.39% (70.38%)**
NO RELATIONSHIP (unigram)	24765	99.71% (92.96%)	99.85% (97.55%)	99.78% (95.20%)
HAS RELATIONSHIP (sentence pair)	**7030**	**97.06% (82.47%)**	**95.89% (63.53%)**	**96.47% (71.77%)**
NO RELATIONSHIP (sentence pair)	48162	99.40% (93.19%)	99.58% (97.36%)	99.49% (95.23%)

Table 2. Best scores and baseline scores from SVM of RC

The baseline F-score for the HAS RELATIONSHIP set of the unigram sentence level is 70.38% and 95.20% in the NO RELATIONSHIP set. The difference can be attributed to the fact that the total number of the NO RELATIONSHIP set is 7 times larger than the HAS RELATIONSHIP set. However, the performance in "has relation" is more important, since the generation of medication entries is based on the pairs which have the relationship correctly identified.

A high performance is achieved in which the F-score for the "has relation" set of the unigram sentence level is 98.39%, while 96.47% is achieved in the bigram sentence level indicating little if any systematic errors.

5.3 CONTEXT Engine Evaluation

The CONTEXT engine was adopted to discover the span of the medication list (the span between the medication heading and the next following heading). The rules which are used in the engine are based on the medication headings in the training set. Table 3 shows the performance of the test set for the CONTEXT engine.

Entity Type	Training	Testing	Recall	Precision	F-Score
Heading pairs	166	51	94.44%	100.00%	97.14%

Table 3. System scores from SVM for determining Context.

An F-score of 97.14% was achieved with the CONTEXT engine.

5.4 Final Output Evaluation

The final evaluation tool used here is released from i2b2 National Center. Due to the errors in the NER, Relationship Classification and Medication Entry Generator, the final F-scores for each entity type are lower than in the NER processing. The final scores for the medication event are between 86.23% and 88.16% (see table 4). The main reason for performance decrease in DOSAGE, MODE, FREQUENCY, DURATION and REASON is because the low recall for the MEDICATION in the NER (computed using CRF). If these medications related entities were extracted without the MEDICATION, these entities could not be connected into medication entries, which make them meaningless in the final output. Another factor is the low performance of REASON extraction by the NER. The frequency of appearance of multiple REASONs is relatively high, and the multiple REASONs should be used to construct multiple medication entries. In this way, the loss in REASON recognition would lead to the decrease in recall of all other entity types and the medication event.

Type	Token Level F-Sore	Entity F-Score
Medication Entry	87.33%	88.16%

Table 4. Final evaluation scores for Medication Entry.

6 Conclusion

In this paper, a high accuracy and comprehensive medication event extraction system is presented. Compared to the three similar systems (see section 2), a better performance is achieved here, even through these systems have a narrower definition for medication event and a different evaluation metric. For example, the F-score of MEDICATION in Sirohi's system is 69.55%, whereas our system achieves 91.40%. As well, the F-score of the exact match for DRUG-DOSAGE event in the Evans's system is 86.76% and 87.92% is obtained in Gold's system for the MEDICATION in their medication event. In contrast, the MEDICATION in the medication event of our system achieves an F-score of 89.16%~90.93%.

In future work, DURATION and REASON are the two main entities that need to be improved. One possible solution is to use the relationship between the medication and its corresponding diseases or symptoms to improve the REASON extraction. As to DURATION, increasing the training set to obtain more examples is probably the best strategy.

Acknowledgments

We would like to acknowledge the contribution of Stephen Crawshaw, Yefeng Wang and other members in the Health Information Technologies Research Laboratory.

Deidentified clinical records used in this research were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY.

References

David A. Evans, Nicholas D. Brownlowt, William R. Hersh, and Emily M. Campbell. 1996. Automating Concept Identification in the Electionic Meidcal Record: An Experiment in Extracting Doseage Information. *AMIA 1996 Symposium Proceedings*, 388-392

Sigfried Gold, Noémie Elhadad. and Xinxin Zhu. 2008. Extracting Structured Medication Event Inforamtion from Dicharge Summaries, *AMIA 2008 Symposium Proceedings*, 237

Jon Patrick, Yefeng Wang and Peter Budd. 2007. An automated system for conversion of clinical notes into SNOMED clinical terminology, *in Proc. 5rd Australasian symposium on ACSW frontiers*, 68: 219-226.

E. Sirohi, and P. Peissig. 2005. Study of Effect of Drug Lexicons on Medication Extraction From Electronic Medical Records. *Pacifi Symposium on Biocomputing.* 10: 308-318

Walter V. Sujansky. 1998. The benefits and challenges of an electronic medical record: much more than a "word-processed" patient chart. *West J Med*, 169(3):176-83.

Improved Text Categorisation for Wikipedia Named Entities

Sam Tardif and **James R. Curran** and **Tara Murphy**
School of Information Technologies
University of Sydney
NSW 2006, Australia
{star4245,james,tm}@it.usyd.edu.au

Abstract

The accuracy of named entity recognition systems relies heavily upon the volume and quality of available training data. Improving the process of automatically producing such training data is an important task, as manual acquisition is both time consuming and expensive. We explore the use of a variety of machine learning algorithms for categorising Wikipedia articles, an initial step in producing the named entity training data. We were able to achieve a categorisation accuracy of 95% F-score over six coarse categories, an improvement of up to 5% F-score over previous methods.

1 Introduction

Named Entity Recognition (NER) is the task of identifying proper nouns, such as location, organisation and personal names, in text. It emerged as a distinct type of information extraction during the sixth Message Understanding Conference (MUC) evaluation in 1995, and was further defined and explored in the CoNLL NER evaluations of 2002 and 2003.

A set of four broad categories became the standard scheme for marking named entities (NEs) in text: person (PER), organisation (ORG), location (LOC), and miscellaneous (MISC). This scheme remains the most common, despite the development of more complex hierarchical category schemes (e.g. Brunstein (2002); Sekine et al. (2002)). Domain-specific category schemes have also been developed in many areas, such as astroinformatics (Murphy et al., 2006), bioinformatics (Kim et al., 2003) and the travel industry (Vijayakrishna and Sobha, 2008). We also extend the broad scheme with a DAB category for Wikipedia "disambiguation" pages — pages used to group articles with identical titles.

NER systems that categorise NEs under these schemes require a large amount of highly accurate training data to perform well at the task. Expert annotation is time consuming and expensive, so there is an imperative to generate this data automatically. Wikipedia is emerging as a significant resource due to its immense size and rich structural information, such as its link structure.

Nothman et al. (2009) introduced a novel approach to exploiting Wikipedia's internal structure to produce training data for NER systems. Their process involved an initial step of categorising all Wikipedia articles using a simple heuristic-based bootstrapping algorithm. Potential NEs were then identified as the words in an article's text that served as links to other Wikipedia articles. To label a NE they then used the category assigned to the article that it linked to.

We have explored the use of Naïve Bayes (NB) and support vector machines (SVMs) as replacements for the text categorisation approach taken by Nothman. This involved the conversion of heuristics used by Nothman into features as well as the incorporation of a number of new features. We demonstrate the superiority of our approach, providing a comparison of the individual text categorisation step to both Nothman's system and other previous research. Our state-of-the-art text categorisation system for Wikipedia achieved an improvement of up to 5% F-score over previous approaches.

2 Background

Accurate classifications for Wikipedia articles are useful for a number of natural language processing (NLP) tasks, such as question answering and NER. To produce article classifications for generating NER training data, Nothman et al. (2009) used a heuristic-based text categorisation system. This involved extracting the first head noun after the copula, head nouns from an article's categories, and incoming link information. They reported an F-score of 89% when evaluating on a set of 1,300 hand-labelled articles.

Dakka and Cucerzan (2008) explored the use of NB and SVM classifiers for categorising Wikipedia. They expanded each article's bag-of-words representation with disambiguated surface forms, as well as terms extracted from its first paragraph, abstract, and any tables present. They also extracted a small amount of context surrounding links to other Wikipedia articles.

Dakka and Cucerzan (2008) expanded their set of 800 hand-labelled articles using a semi-supervised approach, extracting training samples from Wikipedia "List" pages — pages that group other articles by type. For each "List" page containing a link to an article from the hand-labelled set they used the hand-labelled article's category to classify other articles on the list. They neglected to report how many training instances this left them with, but noted that they maintained the original class distribution of the hand-labelled data. They achieved an F-score of 89.7% with an SVM classifier and the category set PER, LOC, ORG, MISC and COM (for common nouns) when classifying their full article set.

We experimented with a combination of the classification techniques used by Dakka and Cucerzan (2008) and the feature extraction methods used by Nothman et al. (2009) and others (Ponzetto and Strube, 2007; Hu et al., 2008; Biadsy et al., 2008), focusing on the extraction of features from Wikipedia's rich metadata.

3 Data

Our annotation and experiments were all run on a March 2009 dump of Wikipedia. The mwlib[1] library was used to parse the Mediawiki markup and perform tasks such as expanding Wikipedia templates and extracting article categories and links. Punkt (Kiss and Strunk, 2006) and the NLTK (Loper and Bird, 2002) were used to tokenise the corpus.

[1] http://code.pediapress.com

New category	Example
PER	
Fictional	Popeye
Animal	Chupacabra
ORG	
Band	Blink-182
LOC	
Geological	Himalayas
MISC	
Franchise	Star Wars
Product → Software	Python

Table 1: Extensions to the BBN categories with examples

3.1 Annotation scheme

Annotation was performed under a slightly modified BBN category hierarchy (Brunstein, 2002). During annotation we discovered the need for a number of additional categories due to the large number of articles Wikipedia contains relating to popular culture, for example the new categories $Organisation \rightarrow Band$ and $Misc \rightarrow Work\ of\ Art \rightarrow TVSeries$ were quite common. We map these categories back to the "Other" subcategory of their parent category to allow accurate comparison with the original BBN scheme. Table 1 lists some of our new categories and gives an example for each.

We also discovered a number of ambiguities in the original BBN scheme. A number of Wikipedia articles were border cases in the BBN scheme — they related to a number of categories, but did not fit perfectly into any single one. The category $Misc \rightarrow Franchise$ is an example of an additional category to label articles such as "Star Wars" and "Final Fantasy". We also noticed some unresolvable overlaps in categories, such as $Location \rightarrow Location \rightarrow Island$ and $Location \rightarrow$ GPE $\rightarrow State$ for articles such as "Tasmania" and "Hawaii".

3.2 Manual annotation

A list of Wikipedia articles was selected for annotation based on several criteria. Given the large number of stub articles that exist within Wikipedia and

the poor representation of categories that selecting random articles would achieve, our list of articles was primarily based on their popularity as detailed by Ringland et al. (2009). We took into consideration the number of different language versions of Wikipedia that the article existed in to try and maximise the usefulness of our annotated data for further multi-lingual NLP tasks. We took a list of the most popular articles from August 2008 and checked for an article's existence on that list. We also considered the number of incoming links an article attracted. Based on these three criteria we produced a list of 2,311 articles for annotation.

Our resulting set of articles was of much higher quality than one that a random article selection process would produce. Random article selection fails to achieve good coverage of some important article categories, such as $Location \rightarrow GPE \rightarrow Country$ which annotators are likely to never come across using a random selection method. Random selection also yields a high number of stub articles with fewer features for a machine learner to learn from.

Our final set of Wikipedia articles was double-annotated with an inter-annotator agreement of 99.7% using the fine-grained category scheme, and an agreement of 99.87% on the broad NER categories. The remaining classification discrepancies were due to fundamental conflicts in the category hierarchy that could not be resolved. This set of hand-labelled articles will be released after publication.

4 Features for text categorisation

Our baseline system used a simple bag-of-words including tokens from the entire article body and the article title. This did not include tokens that appear in templates used in the generation of an article.

We then experimented with a number of different feature extraction methods, focusing primarily on the document structure for identifying useful features. Tokens in the first paragraph were identified by Dakka and Cucerzan (2008) as useful features for a machine learner, an idea stemming from the fact that most human annotators will recognise an article's category after reading just the first paragraph. We extended this idea by also marking the first sentence and title tokens as separate from other tokens, as we found that often the first sentence was all that was required for a human annotator to classify an article. We ran experiments limiting the feature space to these smaller portions of the document.

Wikipedia articles often have a large amount of metadata that helps in identifying an article's category, in particular Wikipedia categories and templates. Wikipedia categories are informal user defined and applied categories, forming a "folksonomy" rather than a strict taxonomy suitable for classification tasks, but the terms in the category names are usually strong indicators of an article's class. We extracted the list of categories applied to each article, tokenised the category names and added each token to the bag-of-words representation of the article.

Using the same reasoning we also extracted a list of each article's templates, tokenised their names, and expanded the article's bag-of-words representation with these tokens. Furthermore, we expanded the templates "Infobox", "Sidebar" and "Taxobox" to extract tokens from their content. These templates often contain a condensed set of important facts relating to the article, and so are powerful additions to the bag-of-words representation of an article. Category, template and infobox features were marked with prefixes to distinguish them from each other and from features extracted from the article body.

We reduced our raw set of features using a stop list of frequent terms, and removing terms with frequency less than 20 in a set of 1,800,800 articles taken from a separate Wikipedia dump. The assumption is that the majority of low frequency tokens will be typographical errors, or otherwise statistically unreliable data.

5 Results

We compared our two classifiers against the heuristic-based system described by Nothman et al. (2009) and the classifiers described by Dakka and Cucerzan (2008). We also tested a baseline system that used a bag-of-words representation of Wikipedia articles with rich metadata excluded. All SVM experiments were run using LIB-SVM (Chang and Lin, 2001) using a linear kernel with parameter $C = 2$. For NB experiments we used the NLTK.

The text categorisation system developed by Nothman et al. (2009) was provided to us by the authors, and we evaluated it using our hand-labelled

training data. Direct comparison with this system was difficult, as it has the ability to mark an article as "unknown" or "conflict" and defer classification. Given that these classifications cannot be considered correct we marked them as classification errors.

There were also a number of complications when comparing our system with the system described by Dakka and Cucerzan (2008): they used a different, and substantially smaller, hand-labelled data set; they did not specify how they handled disambiguation pages; they provided no results for experiments using only hand-labelled data, instead incorporating training data produced via their semi-automated approach into the final results; and they neglected to report the final size of the training data produced by their semi-automated annotation. However, these two systems provided the closest benchmarks for comparison.

We found that across all experiments the NB classifier performed best when using a bag-of-words representation incorporating the first sentence of an article only, along with tokens extracted from categories, templates and infoboxes. Conversely, the SVM classifier performed best using a bag-of-words representation incorporating the entire body of an article, along with category, template and infobox tokens. All experiment results listed were run with these respective configurations.

We evaluated our system on two coarse-grained sets of data: the first containing all articles from our hand-labelled set, and the second containing only those articles that described NEs. Table 2 lists results from the top scoring configurations for both the NB and SVM classifiers. The SVM classifier performed significantly better than the NB classifier.

Limiting the categorisation scheme to NE-only classes improved the classification accuracy for both classifiers, as the difficult NON class was excluded. With this exclusion the NB classifier became much more competitive with the SVM classifier.

Table 3 is a comparison of precision, recall and F-scores between our baseline and final systems, and the systems produced by Nothman et al. (2009) and Dakka and Cucerzan (2008). The difference between results from Nothman's system, our baseline and our full feature classifier were all found to be statistically significant at the $p < 0.05$ level. We performed this significance test using a stratified sampling approach outlined by Chinchor (1992).

(a) Full coarse-grained task

	NB			SVM		
Class	P	R	F	P	R	F
PER	72	98	83	99	92	95
ORG	70	94	80	95	91	93
LOC	97	99	98	99	99	99
MISC	69	84	76	90	88	89
NON	98	57	72	91	96	93
DAB	87	90	88	98	99	98
Micro Avg.	**83**	**83**	**83**	**95**	**95**	**95**

(b) NE-only task

	NB			SVM		
Class	P	R	F	P	R	F
PER	88	98	93	99	94	96
ORG	88	93	90	97	93	95
LOC	99	99	99	99	99	99
MISC	95	85	90	91	97	94
Micro Avg.	**94**	**94**	**94**	**97**	**97**	**97**

Table 2: NB and SVM results on coarse-grained problems.

Classifier	F
Nothman	91
Dakka	90
BASELINE	94
BEST	95

Table 3: Comparison with previous systems.

6 Conclusion

We exploited Wikipedia's rich document structure and content, such as categories, templates and infoboxes, to classify its articles under a categorisation scheme using NB and SVM machine learners. Our system produced state-of-the-art results, achieving an F-score of 95%, an improvement of up to 5% over previous approaches. These high quality classifications are useful for a number of NLP tasks, in particular named entity recognition.

Acknowledgements

We would like to thank the Language Technology Research Group and the anonymous reviewers for their helpful feedback. This work was partially supported by the Capital Markets Cooperative Research Centre Limited.

References

Fadi Biadsy, Julia Hirschberg, and Elena Filatova. An unsupervised approach to biography production using wikipedia. In *Proceedings of ACL-08: HLT*, pages 807–815, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Ada Brunstein. Annotation guidelines for answer types. LDC2005T33, Linguistic Data Consortium, Philadelphia, 2002.

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001.

N Chinchor. Statistical significance of muc-6 results. In *Proceedings, Fourth Message Understanding Conference (MUC-4)*, 1992.

W Dakka and S Cucerzan. Augmenting wikipedia with named entity tags. In *Proceedings of IJC-NLP 2008*, 2008.

Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, Qiang Yang, and Zheng Chen. Enhancing text clustering by leveraging wikipedia semantics. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 179–186, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl. 1):i180–i182, 2003.

Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32:485–525, 2006.

Edward Loper and Steven Bird. NLTK: The natural language toolkit. In *Proceedings of the Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, July 2002.

Tara Murphy, Tara McIntosh, and James R. Curran. Named entity recognition for astronomy literature. In *Proceedings of the Australian Language Technology Workshop*, pages 59–66, Sydney, Australia, 2006.

Joel Nothman, Tara Murphy, and James R. Curran. Analysing Wikipedia and gold-standard corpora for NER training. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 612–620, Athens, Greece, March 2009. Association for Computational Linguistics.

S P Ponzetto and M Strube. Deriving a large scale taxonomy from wikipedia. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 1440–1445, 2007.

Nicky Ringland, James R. Curran, and Tara Murphy. Classifying articles in english and german wikipedias. In *Submitted to ALTA*, 2009.

S Sekine, K Sudo, and C Nobata. Extended named entity hierarchy. In *Proceedings of the LREC-2002*, 2002.

R. Vijayakrishna and L. Sobha. Domain focused named entity recognizer for tamil using conditional random fields. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*, pages 59–66, Hyderabad, India, January 2008.

Towards a flexible platform for voice accent and expression selection on a healthcare robot

Aleksandar Igic[1], Catherine I. Watson[1], Jonathan Teutenberg[2], Rie Tamagawa[3], Bruce MacDonald[1], Elizabeth Broadbent[3]

1 - Department of Electrical and Computer Engineering
2 - Department of Comuter Science
3 - Department of Psychological Medicine

University of Auckland, Private Bag 92019, Auckland 1142,
New Zealand

aigi001@aucklanduni.ac.nz,
jono@cs.auckland.ac.nz,
{c.watson,e.broadbent,r.tamagawa,b.macdonald}
@auckland.ac.nz

Abstract

In the application of robots in healthcare, where there is a requirement to communicate vocally with non-expert users, a capacity to generate intelligible and expressive speech is needed. The Festival Speech Synthesis System is used as a framework for speech generation on our healthcare robot. Expression is added to speech by modifying mean pitch and pitch range parameters of a statistical model distributed with Festival. US and UK English diphone voices are evaluated alongside a newly made New Zealand English accented diphone voice by human judges. Results show judges can discern different accents and correctly identify the nationality of the voice.

1. Introduction

With the rapidly ageing populations in the developed world, robots are increasingly finding a use in nursing homes in assistive medical care [1][2]. In order for such robots to facilitate the needs of older and mobility restricted users from a communication point of view, more human modes of interaction need to be implemented [1]. The most natural mode of communication for humans is speech, which for a medical robot requires both speech recognition and generation capabilities. We are currently focusing on implementing a flexible robotic speech generation framework that will provide a high standard of quality and expressiveness.

2. Healthcare Robot Background

We are currently developing a Healthcare robot to assist with elderly care. This multi-disciplined project involves personnel with backgrounds in Engineering, Health Psychology, Health Informatics, Nursing, and Gerontology. It involves academics and industry from both New Zealand and Korea [3] .The project is working closely with a retirement village in Auckland, where the healthcare robots are to be trialed. We have already evaluated a preliminary version of the robot with elderly users, in a blood pressure measurement task. The robot instructed users how to use a blood pressure measurement device, and reported back their measurements [4]. The robot (see *Figure 1*) is a mobile device with ultra sound and laser sensors for location detection. It has a screen with a talking virtual head and the face is able to convey a variety of emotions [5]. We discuss the development of the expressive face, accompanying the speech synthesis in [4].

Figure 1: *Charles with blood pressure monitor.*

The functionality of the healthcare robot is now being extended to include location monitoring, falls detection, medication management, appointment reminders, and more vital signs measurements (pulse and blood oxygenation) [1][6]. Some other non medical uses could include, delivering weather and time information and reading the news. These roles require, in most cases, human robot interaction to take place in form of speech dialogue.

The robot voice is provided by the Festival Speech Synthesis system [7]. The preliminary version of the robot used one of the default voices (KAL), which was male with an American accent. Feedback from the preliminary study [5] revealed that users found the voice the voice "too robotic". To this end we have been investigating a variety of ways to make the voice more engaging. We have considered using different accented voices and different models of intonation, and have implemented simple emotion models, and a more flexible speech synthesis system. This paper outlines our development of creating different voices for our robot, and presents evaluations of the voices to date.

3. Speech synthesis

Festival offers a robust and flexible architecture for speech and language modeling, with a powerful capability to easily integrate new speech generation modules. Scripting functionality is implemented through an internal Scheme interpreter. The standard Festival distribution contains automatic intonation and duration generating schemes, as well as a facility for manual intonation modeling through ToBI [8]. Speech synthesis methods in Festival include: Diphone concatenation [9], Multisyn unit selection [10] and HTS hidden Markov synthesis [11]. Festival is implemented on the robot in server mode, and interacts with the rest of the robot modules through a modified Player [12] framework.

In our studies we have used the three differently accented English synthetic voices generated through diphone concatenation: US, UK and NZ. US English and UK English are the two diphone voices that are part of the standard Festival distribution. The NZ voice is newly developed at the University of Auckland and contains diphones recorded by a male speaker and a New Zealand English lexicon with 500 common Maori words [13].

4. Adding Expression to Speech

We have subdivided the robot dialogue into five different types: greeting, instruction (eg. instructing a patient to put a cuff for blood pressure measurements), information (eg. delivering measurement results), question, social (eg. reading news, telling jokes). Our goal is to ensure that each of these dialogue types has the appropriate tone. This means we need to be able to adjust the both the intonation and emotion quotient of the voice. At present we are only employing very simple techniques, but coupled with the virtual robot head we can convey different emotional states.

Generating expressive intonation is a multi-tier process within Festival. The text to be spoken is first ToBI labeled [8] manually, or automatically through a CART tree model [14] [15]. These labels are then converted to pitch targets using linear regression [16]. Interpolation is done between target points to generate a pitch contour for the utterance. Two parameters, mean speaker pitch and speaker pitch standard deviation, allow control over the average value and the range of the final pitch contour.

4.1 A New Method of Changing Intonation in Festival

To allow for utterances to be synthesized with different levels of expression, a function 'SayEmotional' was written in Scheme which takes three parameters: input text to be synthesized, one of two emotions 'Happy' or 'Neutral', and the level of 'emotional intensity'. *Figure 2* illustrates this functionality by comparing plots of pitch contours of the utterance "I am very happy to meet you" generated with four different methods: *a)* with no intonation, *b)* with manually labeled text, *c)* and *d)* through 'SayEmotional' utilizing 'Neutral' and 'Happy' parameters respectfully.

These plots show the value of the fundamental frequency (f0) of voiced speech as it changes throughout the duration of the utterance. All are generated with the New Zealand English voice, and are of the same duration.

The 'Happy' utterance differs from the 'Neutral' through increased pitch mean and range. This follows findings in psychological studies of acoustic properties of emotion as reviewed in [17].

The case of no intonation being applied the contour is flat, and in the manually labeled text case:

(I((accent H*))) (am((accent L*))) (very((accent H*)))(happy((accent L*))) (to()) (meet((accent H*))) (you((tone L-H%)))

The contour is dynamic, with f0 rises occurring at (H*) labeled words, and f0 falls occurring at (L*) labeled words.

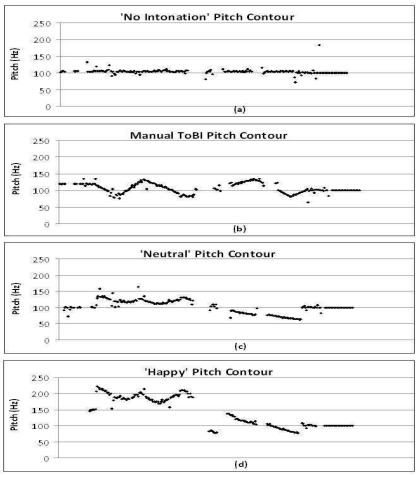

Figure 2: *Pitch contours of the phrase "I am very happy to meet you" with: a) no intonation, b) manual ToBI intonation, c) automatically generated 'Neutral' intonation and d) automatically generated 'Happy' intonation*

4.2 Changing Emotion

The 'SayEmotional' method makes use of the automatic intonation generation and manipulates the two baseline linear regression parameters to generate emotional speech. The baseline parameters are mean pitch and pitch standard deviation, and they are calculated from original recorded diphones. These parameters are dependent on the vocal characteristics of the speaker and are different for each diphone voice. As the diphones are context neutral, baseline parameters for mean pitch and range are used for generating 'Neutral' utterances.

In order to vary the emotive state of the generated speech, we are systematically changing the mean and the standard deviation parameters of the CART model. To move from 'Neutral' to lowest intensity 'Happy' we are increasing the mean pitch to 1.5 times and the standard deviation to 2 times that of the original. High intensity 'Happy' is achieved by increasing the mean to 2.5 and standard deviation to 4 times that of the original.

5. Diphone Voice Evaluations

A study was conducted to evaluate the human perception on the three English accented voices: US, UK and NZ. All voices are synthesized using diphone concatenation using context neutral diphones. The study group comprised of 20 participants, 6 males and 14 females with a mean age of 31.95 and standard deviation of 11.65. Participants had lived in NZ for average of 20.87 years with standard deviation of 12.26.

In the procedure each participant was asked to listen to a minute long sentence synthesized by one of the three English voices. Two sentences were synthesized per voice; one with manually and the other with the automatically ToBI annotated text, comprising in total of 6 different sentences being evaluated by each participant. Three measures were investigated: the quality of the voice, the nationality of the voice and the 'roboticness' of the voice, lastly participants were asked to indicate which voice was the most preferred and which was the least.

ANOVA analysis was performed on all the results and showed no significant differences in the quality score among the voices regardless of whether the intonation of the speech was generated from ToBI labels, automatically generated or labeled by hand., $F(5, 114) = 1.75$, $p = .128$.

When participants were asked to rate the roboticness of the voice, the results of ANOVA showed that the rating was significantly different between the 6 voices, $F(5, 114) = 2.31$, $p = .048$. The US original voice was rated as the most robotic while NZ original was rated as the most human-like. There was no significant difference in roboticness between intonation from automatically generated ToBI labels or labeled by hand. Since there was no difference in quality and roboticness between the two intonation methods, we will focus on the results of the intonation from the automatically generated ToBI labels from now on.

We tested to see whether the participants could identify the accent type of the voices. Each was given 9 options (New Zealand, Australian, South African, British, Asian, Canadian, American, Irish, Other (non-definable)). The majority of participants guessed the correct nationality of the given voice although the recognition rate for US voices was lower than for NZ and UK voices.

The New Zealand accent was correctly identified by 65% of the participants, the US accent was correctly identified by 45 % of the participants, and the UK accent by 50 % of the participants.

Preferred		Non preferred	
	% recalled		% recalled
New Zealand	35	New Zealand	35
American	10	American	55
British	55	British	10

Table 1: *Preferred and non preferred accent*

We also asked the participants what accent they preferred the most and the least (see table 1). Results of Chi-square were significant for both Preferred $\chi^2(2, N = 20)= 6.10, p = .047$, and Non-preferred answers, $\chi^2(2, N = 20)= 6.10, p = .047$, indicating that participants had significantly varied opinions about which voice they prefer and do not prefer. The British accent was preferred by more participants, while American accent was least preferred.

The main outcome of the study shows that there is no statistical difference between the effects of manual and automatic intonation schemes on the perception of quality and the roboticness of synthetic voices. Due to these results we have decided to move away from manual ToBI labeling and focus solely on automatic intonation schemes. This realization in turn prompted the development of the 'SayEmotional' method described in Section 4, which was based solely on adapting the automated intonation scheme. These results also indicate that there is a personal preference element in voice accent. This is suggests users should have a choice about the voice nationality on the robot.

6. Improving the speech synthesizer

Manipulating the pre-recorded diphone speech waveforms through intonation modeling as with 'SayEmotional' introduces audible artifacts that reduce the quality of the generated speech. Currently a harmonic plus noise synthesis model (HNM) is being added into Festival which allows for waveform manipulation to be achieved with a lower loss of quality compared to other systems [18]. We have further improved the original HNM system described in [18] by using continuous sinusoids to synthesize speech [19] which further improves the quality of generated speech and gives a two fold increase in computational efficiency of generating speech. The initial focus of the work is to allow New Zealand diphone voice synthesis to work with the HNM system. Eventually we aim to incorporate it into other synthesis methods within festival.

7. Conclusion

We are working on a healthcare robot for nursing homes, with a flexible speech synthesis system as a means of human robot interaction. In the final stage, we intend to have a speech framework with the ability to automatically generate emotive, high quality speech with the capacity to change the nationality of the voice dependant on user preference. The speech system, based on Festival, makes use of differently accented voices including a newly created New Zealand English voice. It is able to change its speech emotive state depending on the context. We are in the process of implementing an improved harmonic plus noise model of speech synthesis.

Throughout the development, there will be usability trials. Next trials, scheduled for October, will focus on the interactions of older people with the robot system in a nursing home.

Acknowledgement

This work was supported by the R&D program of the Korea Ministry of Knowledge and Economy (MKE) and the Korea Evaluation Institute of Industrial Technology (KEIT). [2008-F039-01, Development of Mediated Interface Technology for HRI].

This work is supported by a grant from the NZ government Foundation for Research, Science and Technology for robotics to help care for older people and by a University of Auckland New Staff Grant.

Authors would like to acknowledge the work of Xingyan Li and Tony Kuo, for their work on the development of the Healthcare Robot.

References

[1] I. H. Kuo, E. Broadbent, B. MacDonald. *"Designing a robotic assistant for healthcare applications"*, in the 7th conference of Health Informatics New Zealand, Rotorua, Oct 2008.

[2] J.C. Bauer *"Service robots in health care: The evolution of mechanical solutions to human resource problems"*, Bon Secours Health System, Inc. Technology Early Warning System – White Paper. 2003. Available from: http://bshsi.com/tews/docs/TEWS%20Service%20Robots.pdf.

[3] B. MacDonald, W. Abdulla, E. Broadbent, M. Connolly, K. Day, N. Kerse, M. Neve, J. Warren, C. I. Watson *"Robot assistant for care of older people"*, 5th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2008) November 20-22, 2008.

[4] X. Li, B. MacDonald, C. I. Watson *"Expressive Facial Speech Synthesis on a Robotic Platform"*, 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS 2009, St Loius, October 11 to 15.

[5] I. H. Kuo, J. M. Rabindran, E. Broadbent, Y. I. Lee, N. Kerse, R. M. Q. Stafford, B. MacDonald, *"Age and gender factors in user acceptance of healthcare robots"*. In Proceedings of 18th IEEE International Symposium on Robot and Human Interactive Communication, Toyama, Japan, September 27 – October 2 2009, 214-219.

[6] E. Broadbent, R. Tamagawa, N. Kerse, B. Knock, A. Patience, B. MacDonald "*Retirement home staff and residents' preferences for healthcare robots"*, 18th IEEE International Symposium on Robot and Human Interactive Communication, Toyama, Japan, September 27-October 2, 2009, 645 - 650.

[7] A. W. Black, P. Taylor, R. Caley, *"The Festival Speech Synthesis System"*. In http://www.cstr.ed.ac.uk/projects/festival/, 1999

[8] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, J. Hirschberg, *"TOBI: a standard for labeling English prosody"*. In Second International Conference on Spoken Language Processing, October 13 - 16, 1992, 867-870.

[9] F. Charpentier, M. Stella, *"Diphone synthesis using an overlap-add technique for speech waveforms concatenation"*. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP, April 1986, 2015-2018.

[10] R. A. J. Clark, K. Richmond, S. King, *"Multisyn: Open-domain unit selection for the Festival speech synthesis system"*. In Speech Communication, Volume 49 Issue 4, 2007, 317-330.

[11] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, K. Tokuda *"The HMM-basedSpeech SynthesisSystem (HTS) Version 2.0"*. In Proc. of Sixth ISCA Workshop on Speech Synthesis, 2007

[12] B. P. Gerkey, R. T. Vaughan, A. Howard *"The player/stage project: Tools for multi-robot and distributed sensor systems"* In Proceedings of the International Conference on Advanced Robotics, June 30 –July 3, 2003, 317-323.

[13] C. I. Watson, J. Teutenberg, L. Thompson, S. Roehling, A. Igic, *"How to build a New Zealand voice"*, (Submitted), NZ Linguistic Society Conference, Palmerston North, November 30 – December 1, 2009.

[14] A. K. Syrdal, J. Hirschberg, J. McGory, M. Beckman, *"Automatic ToBI prediction and alignment to speed manual labeling of prosody"* In Speech Communication, Volume 33, Issues 1 - 2, 2001. 135-151.

[15] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, *"Classification and Regression Trees"* Chapman & Hall (Wadsworth, Inc.): New York, 1984.

[16] A. W. Black, A. J. Hunt, *"Generating F0 contours from ToBI labels using linear regression"*. In Fourth International Conference on Spoken Language Processing, October 3 – 6, 1996, 1385-1388.

[17] K. R. Scherer, *"Vocal communication of emotion: A review of research paradigms"*. In Speech Communication, Volume 40, Issue 3, April 2003, 227 – 256.

[18] Y. Stylianou, *"On the implementation of the harmonic plus noise model for concatenative speech synthesis"*, In Third ESCA/COCOSDA Workshop on Speech Synthesis, November 26-29, 1996, 261-266.

[19] J. Teutenberg, C. I. Watson, *"Flexible and efficient harmonic resynthesis by modulated sinusoids*". In the Proceedings of the 17th European Signal Processing Conference, Glasgow, August, 2009.

Integrating Verb-Particle Constructions into CCG Parsing

James W. D. Constable and **James R. Curran**
School of Information Technologies
University of Sydney
NSW 2006, Australia
{jcon6353,james}@it.usyd.edu.au

Abstract

Despite their prevalence in the English language, multiword expressions like verb-particle constructions (VPCs) are often poorly handled by NLP systems. This problem is partly due to inadequacies in existing corpora; the primary corpus for CCG-oriented work, CCGbank, does not account for VPCs at all, and is inconsistent in its handling of them. In this paper, we apply some corrective transformations to CCGbank, and then use it to retrain an augmented version of the Clark and Curran CCG parser. Using our technique, we observe no significant change in F-score, while the resulting parse is semantically more sound.

1 Introduction

Multiword expressions (MWEs), compound lexemes made up of two or more words that together form a complete semantic unit, are one of the problems facing natural language processing systems. Verb-particle constructions (VPCs) are a common type of MWE, comprising a verb and a particle, most often a preposition. The meaning of some VPCs can be logically attributed to the component parts (e.g., *picked out*), but many are idiomatic and semantically opaque (e.g., *make out*).

Previous research into VPCs has focussed much attention on their automatic extraction and classification (Baldwin and Villavicencio, 2002; Villavicencio, 2003). However, research into how they should be handled by parsers is noticeably lacking. Their unusual ability to manifest in both a 'joined' and 'split' configuration ('*gunned down the man*' versus '*gunned the man down*') prevents parsers from treating them as a single unit, and demands a system that is able to maintain the semantic bond between the components, even when they are non-adjacent.

To compound the problem, existing corpora are not consistent in their handling of these constructions. The Penn Treebank (Marcus et al., 1993, 1994) has an RP tag for particles, but sometimes labels them as adverbs. The CCGbank (Hockenmaier and Steedman, 2007) analysis of particles varies, but leans towards treating all particles as adverbial modifiers. This is in itself problematic, since it fails to take into account the fact that particles are a core part of the construct, whereas adverbs are optional. This lack of quality corpora for VPC-related work limits the power of corpus-trained parsers.

In this paper we draw on the Penn Treebank and PropBank (Kingsbury and Palmer, 2003) to repair CCGbank's representation of VPCs, and demonstrate how our approach is able to satisfactorily account for most VPC-related phenomena. Retraining the Clark and Curran parser (Clark and Curran, 2007) on our modified corpus, we observe a very slight decrease in parser F-score, although this is balanced by the fact that the parses now make structural sense.

2 Combinatory Categorial Grammar

Combinatory Categorial Grammar (CCG, Steedman (2000)) is a lexicalised grammar formalism based on combinatory logic. One of the features that makes CCG so appealing to NLP researchers is its high degree of *lexicalisation* (i.e., the degree to which the grammar is built into the lexicon). Every word is assigned a category, and parsing is simply a matter of finding the right sequence of combinators to form a sentence. Recent work has seen the creation of high-performance parsers built on the CCG formalism (Clark and Curran, 2007).

The primary corpus for CCG-related work is CCGbank — an augmented version of the Penn Treebank (Marcus et al., 1993) that contains CCG derivations and predicate argument structures. It was induced from the Penn Treebank (Hocken-

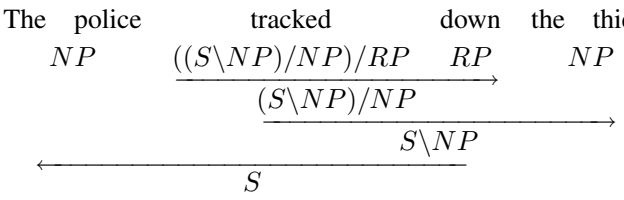

Figure 1: The default case — a VPC in the joined configuration.

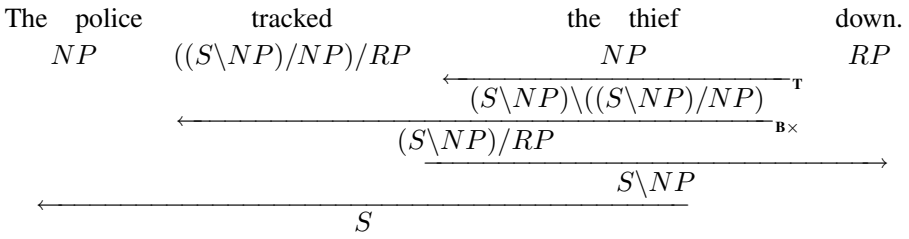

Figure 2: Using type-raising and backward-crossed-composition to handle the split configuration.

maier, 2003) with the goal of furthering CCG research by providing a large corpus of suitably annotated data.

Despite its utility, CCGbank is not without its flaws. Hockenmaier explains that the Treebank does not contain enough information to perform a perfect automatic translation to CCG, and points to complement/adjunct distinctions, phrasal verbs and compound nouns as problematic areas. Some attempts have been made to rectify this; for example, Honnibal and Curran (2007) target the complement/adjunct distinction. Most relevant to our work is the failure to capture phrasal verbs, resulting in the unfortunate situation of particles being treated as adverbial modifiers, and verbs failing to subcategorise for them.

3 CCG Representation of VPCs

Before modifying CCGbank, we first had to determine a suitable method of representing VPCs in CCG. The representation would ideally minimise the ambiguity of the lexical categories, and maintain CCG's transparent interface between syntax and semantics.

The current representation in CCGbank tends to favour an adverb-style treatment, where the verb is assigned a normal verbal category, and the particle is given the category $(S\backslash NP)\backslash(S\backslash NP)$ (i.e. a post-modifying adverb). This approach is semantically rather unsatisfying. The particle in a VPC is not an optional modifier, but a fundamental and obligatory part of the construction. Consider the VPC *gun down* ('to shoot someone or something so that they fall'), and the raw verb *gun* ('to rev up an engine'); clearly the particle is playing much more than a modifying role.

A better approach would be to make the particle a required part of the construction by building it directly into the verb's subcategorisation frame. In the preceding example, we could conceive the VPC version of the verb to have the category $((S\backslash NP)/NP)/Particle$. The question is then how the particle should be represented. None of the existing atomic categories (N, NP, S, PP) work well in this situation, and all open the door to CCG transformations that would be undesirable in this context. Consequently, we chose to introduce a new tag, RP.

In the simplest case, we have the joined configuration (shown in Figure 1), which requires only functional application. The joined configuration was chosen as the default due to its overwhelming prevalence.

The rarer split case (shown in Figure 2) is slightly more complicated. We use a combination of type-raising and backward-crossed-composition (similar to the Steedman and Baldridge (2006) analysis of heavy noun phrase shift), whilst leaving the verb and particle categories unchanged.

An alternative option for the split case would have been to simply introduce a new category for the verb. However, this approach increases the category ambiguity of the words, and is also opposed to the general design of the formalism, which prefers to handle such surface variation using only combinatory rules.

Finally, we show that our representation can

comfortably accommodate a coordination construction where two verbs share a particle. This relatively rare particle sharing phenomenon occurs only once in the Penn Treebank, and is dealt with in our representation using the simplified coordination combinator, as shown in Figure 3.

One problem with the representation is its tendency to over-generate. English grammar requires that VPCs with a pronominal object be in the split configuration (*she took it away* but not **she took away it*), however this restriction is not observed in our representation, thus allowing invalid sentences. The same applies for manner adverbs occurring between the verb and the particle; English grammar disallows constructs like **they tracked quickly down the thief*, however these are accepted in our proposed representation.

4 Modifying the Corpus

The next stage in our process was modifying CCGbank to accommodate the changes. This involved changing both the syntactic derivations and the word-word dependencies in the predicate-argument structure. The details of the structure of CCGbank can be found in Hockenmaier and Steedman (2005).

To simplify the manipulation of the CCG structures, we first read them into Python as tree-structures, and then wrote these to an external database[1]. This gave us quite a lot of flexibility in querying, retrieving and modifying the structures.

To locate VPCs within the corpus, we relied on a combination of PropBank's (Kingsbury and Palmer, 2002) argument structure labeling and the tags in the Penn Treebank. PropBank provides a listing of every verb (relation) in the corpus, along with its arguments. The word positions for each relation and its arguments are also given, making multiword relations (such as VPCs) readily identifiable. Whenever a multiword relation was found that also contained an RP tag in the Penn Treebank (RP being the Penn Treebank's tag for particles), we took that set of words as being a VPC. This approach errs on the side of caution — there are some valid VPCs in the Penn Treebank that do not have the particle tagged as RP.

A quick survey of the discovered VPCs revealed some interesting features. In total there were 2,578 VPCs. Grouping them based on whether or not the verb and particle share the same parent node in the CCGbank derivation (which loosely equates to the joined-split distinction) yields the results in Table 2. Such a decisive split indicates that there is a definite bias towards the joined configuration, which has the advantage of simplifying the common joined case, but making the split cases even more difficult to identify.

	Same Parent	Different Parents
Count	2425	153
%	94.1%	5.9%

Table 2: Verb and Particle parents in CCGbank

Count	Category
1339	(S\NP)/NP
647	S\NP
302	(S\NP)/PP
96	((S\NP)/PP)/NP
89	(S\NP)/(S\NP)
69	(S\NP)/S
15	((S\NP)/(S\NP))/NP
4	((S\NP)/NP)/PP
3	((S\NP)/PP)/PP
3	N
2	(S\S)\NP

Table 3: Summary of the Verb Categories.

Count	Category
2541	(S\NP)\(S\NP)
10	PP/PP
8	PP/NP
5	(S\NP)/PP
3	S\S
3	N\N
2	((S\NP)\(S\NP))/PP
2	S\NP

Table 4: Summary of the Particle Categories.

Tables 3 and 4 summarise the original CCG categories assigned to the verbs and particles in each VPC that occurred more than once. There is a lot of variation in the tail of each distribution as well as several erroneous categories, although both groups have one category that clearly dominates the rest. The verbs are dominated by the transitive and intransitive categories and the particles are almost exclusively tagged as adverbial modifiers.

For each of the main categories assigned, we hand-crafted a transformation rule to convert instances of that category to our CCG representa-

[1] Acknowledgements to Tim Dawborn for his preparatory work on this system.

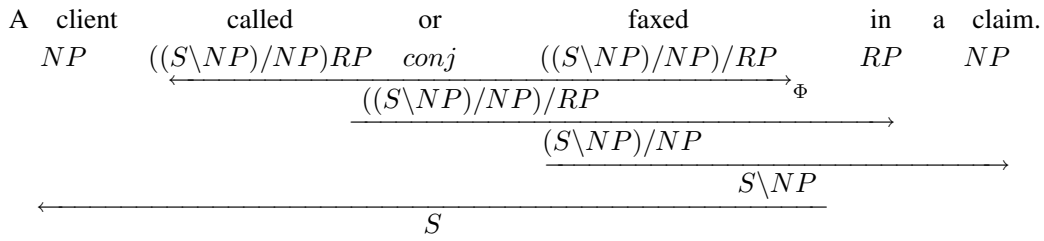

Figure 3: Using the coordination combinator to handle shared particles

Model	LP	LR	LF	LF (POS)	SENT ACC	UP	UR	UF	CAT ACC	COV
C&C	88.06	86.43	87.24	85.25	35.67	93.88	92.13	93.00	94.16	99.06
C&C + VPC	87.90	86.34	87.11	85.11	35.73	93.80	92.13	92.96	94.06	99.06

Table 1: Comparison of results before (top line) and after (bottom line) using the modified VPC corpus.

tion. Instances of VPC nominalisation and categories that occurred with low frequently were left untouched (about 25 instances in total).

5 Results

After modifying the Clark and Curran parser to include support for the new categories that were produced by the conversion process, we retrained the parser on the modified corpus, and then retested it using the same procedure described in Clark and Curran (2007). Our results are shown in Table 1, along with those obtained by Clark and Curran on the unmodified corpus using their hybrid model.

The LP, LR, and LF columns give the labelled precision, recall and F-score respectively for labelled CCG dependencies. We can see that there was a very slight decrease in performance, however considering that the task has been made more difficult by the addition of categories and the resulting parse is structurally and semantically more sound, this is a very small penalty. The statistics for the unlabelled dependencies (UP, UR and UF) show a similar trend. Additionally, as 5.09% of the sentences in the corpus contained VPCs (using our method of detection), we could assume that consistent misclassification would have led to a much larger performance hit.

Table 1 also shows the labelled F-score on automatically assigned POS tags, which also has a similar small performance drop. This is surprising because we expected the preposition/particle distinction to be more challenging for the POS tagger, and that these errors would flow onto the parser.

Table 5 shows the performance of the verb-particle dependencies themselves. There are 97 VPCs in Section 00, and the parser successfully retrieves the vast majority of them, even with automatically assigned POS tags. However, it is far worse at correctly determining the full subcategorisation frame for the verbs, with only 67% of verb categories (65 of 97) being completely correct with gold POS tags.

Type	Frequency
in Gold Standard	97
found by parser (gold POS)	96
found by parser (auto POS)	91
given correct category (gold POS)	65
given correct category (auto POS)	56

Table 5: VPCs in CCGbank Section 00

6 Conclusion

By employing both PropBank and the Penn Treebank, we have been able to produce a modified version of the CCGbank corpus that contains a more syntactically and semantically sound annotation of VPCs. Training the Clark and Curran CCG parser on the new corpus produced equivalent empirical results to the original parser, despite the additional complexity of the augmented corpus. Our initial results demonstrate that VPCs can be parsed efficiently and in a linguistically sophisticated manner using CCG.

7 Acknowledgements

This work was partially supported by the Capital Markets Cooperative Research Centre Limited. We would also like to thank the anonymous reviewers for their helpful comments.

References

Timothy Baldwin and Aline Villavicencio. Extracting the unextractable: a case study on verb-particles. In *Proceedings of the 2002 Conference on Natural Language Learning*, pages 1–7, Taipei, Taiwan, August 2002.

Stephen Clark and James R Curran. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics*, 33(4):493–552, 2007.

Julia Hockenmaier. Data and models for statistical parsing with Combinatory Categorial Grammar. *School of Informatics, Edinburgh, University of Edinburgh*, 280, 2003.

Julia Hockenmaier and Mark Steedman. CCGbank: Users' manual. *Technical Reports (CIS)*, page 52, 2005.

Julia Hockenmaier and Mark Steedman. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396, 2007.

Matthew Honnibal and James R Curran. Improving the complement/adjunct distinction in CCGBank. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING-07)*, pages 210–217, 2007.

Paul Kingsbury and Martha Palmer. From treebank to propbank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, pages 1989–1993, 2002.

Paul Kingsbury and Martha Palmer. Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, 2003.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and B Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Mitchell P Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The Penn Treebank: annotating predicate argument structure. In *HLT '94: Proceedings of the workshop on Human Language Technology*, pages 114–119, Morristown, NJ, USA, 1994. Association for Computational Linguistics. ISBN 1-55860-357-3.

Mark Steedman. *The Syntactic Process*. Massachusetts Institute of Technology, USA, 2000.

Mark Steedman and Jason Baldridge. Combinatory Categorial Grammar. *Encyclopedia of Language and Linguistics*, 2:610–622, 2006.

Aline Villavicencio. Verb-particle constructions and lexical resources. In *Proceedings of the Meeting of the Asociation for Computational Linguistics: 2003 workshop on Multiword expressions*, pages 57–64, Sapporo, Japan, July 2003.

From Lexical Entailment to Recognizing Textual Entailment Using Linguistic Resources

Bahadorreza Ofoghi
Centre for Informatics & Applied Optimization, University of Ballarat
Victoria 3350, Australia
b.ofoghi@ballarat.edu.au

John Yearwood
Centre for Informatics & Applied Optimization, University of Ballarat
Victoria 3350, Australia
j.yearwood@ballarat.edu.au

Abstract

In this paper, we introduce our *Recognizing Textual Entailment* (RTE) system developed on the basis of Lexical Entailment between two text excerpts, namely the *hypothesis* and the *text*. To extract atomic parts of hypotheses and texts, we carry out syntactic parsing on the sentences. We then utilize WordNet and FrameNet lexical resources for estimating lexical coverage of the text on the hypothesis. We report the results of our RTE runs on the Text Analysis Conference RTE datasets. Using a failure analysis process, we also show that the main difficulty of our RTE system relates to the underlying difficulty of syntactic analysis of sentences.

1 Introduction

Success in many automated natural language applications implies an accurate understanding of the meaning (semantics) of texts underlying the surface structures (syntax) by machines. This becomes challenging with different syntactic forms and dissimilar terms and phrases expressing the same semantics. Automated natural language applications make extensive use of fine-grained text processing modules that enable them in more effective dealings with structurally complicated texts.

One of the current text processing tasks is concerned with inferring the meaning of a piece of text from that of another potentially larger text excerpt. This has now become a direction of study for the members of the natural language processing community and is known as *Recognizing Textual Entailment* (RTE). The problem of RTE is formally described as recognizing the relationship between a pair of texts referred to as *hypothesis* and *text*. The hypothesis (H) is a succinct piece of text and the text (T) includes a few sentences the meaning of which may or may not entail the meaning of the hypothesis. If the meaning of H can be inferred from that of T, then the relationship is denoted by $T \rightarrow H$. For instance, given H="*UN peacekeepers abuse children.*" and T="*Children as young as six are being sexually abused by UN peacekeepers and aid workers, says a leading UK charity.*" the relation $T \rightarrow H$ holds true.

The classification of the relationship between the hypothesis and the text can be either a 3-way classification or a 2-way classification task. The 3-way classes are:

- *Entailment*: where $T \rightarrow H$.
- *Contradiction*: where $T \rightarrow \neg H$.
- *Unknown*: where there is not enough evidence available in the text to decide whether $T \rightarrow H$ or $T \rightarrow \neg H$.

In the 2-way classification method, the *Contradiction* and *Unknown* relations are unified into a single class called *No Entailment*. Our RTE system only considers the 2-way classification task.

2 Related work

A few approaches to RTE have been developed during recent years. This includes the following.

Term-based approach – Most of the systems that take this approach consider morphological and lexical variations of the terms in texts and hypotheses

and determine the existence of entailment between the texts and hypotheses by means of their lexical similarities (Braz et al., 2005; Pazienza et al., 2005; Rodrigo et al., 2008).

Logic-proving approach – The systems that follow this approach apply elements of classical or plausible logic to infer whether the meaning of the text entails that of the hypothesis. The logical procedures are called on a number of feature elements of the texts and hypotheses such as propositions or other logic forms (Akhmatova and Molla, 2006; Tatu and Moldovan, 2005; Clark and Harrison, 2008).

Syntax-based approach – Some existing systems carry out a similarity analysis between the dependency trees extracted from the texts and hypotheses in order to identify the entailment relationships (Lin and Pantel, 2001; Kouylekov and Magnini, 2005; Yatbaz, 2008). There are also systems that take a *paraphrase detection* strategy to generate a set of different styles of the hypotheses with the aim of searching for a subset of which may occur in the texts (Bosma and Callison-Burch, 2006).

Semantic role-based approach – There are systems that annotate the sentences of the texts and hypotheses with semantic roles (using *shallow semantic parsers*) and then analyze the coincidences between sets of assigned semantic roles (Braz et al., 2005).

Knowledge-based approach – The utilization of world knowledge in these systems facilitates recognizing entailment relationships where existing lexical or semantic knowledge is not adequate for confidently inferring the relationships. One available structure that is moving towards formulating world knowledge is Cyc[1]. We have not found any previous RTE system that uses Cyc.

Our RTE system takes the term-based (lexical) approach to make decisions about textual entailment relationships.

3 System architecture

3.1 Preprocessing and sentence extraction

The preprocessing stage is necessary in order for sentence extraction and the syntactic analysis of the sentences to be successfully carried out. Our RTE system performs some basic grammatical and punctuation fixes, such as adding a "." at the end of sentences if the "." is missing or capitalizing the first letter of a sentence if necessary.

We utilize the *LingPipe*[2] sentence splitter to extract sentences from hypotheses and texts.

3.2 Proposition extraction

Propositions are extracted from each sentence in the hypothesis and the text. A proposition is an atomic representation of concepts in the texts in which there are no clauses or dependent parts of texts included. For instance, from the sentence "*The girl playing tennis is not my friend.*" the proposition "*girl playing tennis*" can be extracted.

Table 1: New syntactic rules for extracting propositions

Linkage	Elements
AN-Mg	AN: connects noun modifiers to nouns, Mg: connects certain prepositions to nouns
AN-Ss/Sp-MVp-Js/Jp	S_: connects subjects to verbs, MVp: connects prepositions to verbs, J_: connects prepositions to their objects
Ss/Spx-Pg*b-Pv-MVp-Js/Jp	Pg*b: connects verbs to present participles, Pv: connects forms of "be" to passive participles

To extract propositions, we use *Link Grammar Parser* (LGP) (Sleator and Temperley, 1993) and follow the procedure explained in (Akhmatova and Molla, 2006). There are seven rules introduced in (Akhmatova and Molla, 2006) and three new rules that we have developed for extracting propositions. Table 1 shows our new syntactic rules. Given the sentence "*Children are being sexually abused by peacekeepers.*", for instance, the output parse will be like what is shown in Figure 1. From this, we are able to extract the proposition "*peacekeepers abuse children.*".

3.3 Lemmatization

Before semantic alignment is carried out, all hypothesis and text terms are lemmatized using *TreeTagger* (Schmid, 1994). This means that the terms are unified to their single lemma like the transformation of the terms "*abusing*" and "*abused*" to the lemma "*abuse*".

[1] http://www.cyc.com/

[2] Alias-i. 2008. LingPipe 3.8.2. http://alias-i.com/lingpipe.

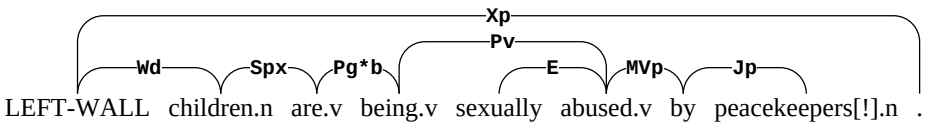

Figure 1: LGP output of the sentence "*Children are being sexually abused by peacekeepers.*"

3.4 Entailment checking

We finally check the entailment between each pair of propositions extracted from the hypothesis and the text. The idea here is that the truth of each single proposition in the hypothesis needs to be entailed at least by the meaning of a proposition in the text in order for our RTE system to decide whether the text entails the truth of the hypothesis.

Checking the pairwise entailment between propositions in our work focuses on the lexical items occurring in the propositions. At this stage, we find the relationships between pairs of lexical items in the propositions regardless of their position. If all lexical items of the hypothesis proposition have related terms in the text proposition, then the decision is that the hypothesis proposition is entailed by the text proposition and an *Entailment* relation is assigned to the pair; otherwise, a *No Entailment* relation is assigned to the hypothesis-text pair.

We use two lexical resources, WordNet (Miller et al., 1990) and FrameNet (Baker et al., 1998), to find relationships between different lexical items. When using WordNet, we assume that a term is semantically interchangeable with its *exact occurrence*, its *synonyms*, and its *hypernyms*. In extracting hypernyms, we only traverse the path in the corresponding WordNet synset for two links.

In utilizing FrameNet, if two lexical items are covered in a single FrameNet frame, then the two items are treated as semantically related in our work. The two verbs "*fly*" and "*pace*", for instance, are covered in (inherited from) the same FrameNet frame "*Self_motion*"; therefore, we assume that these two verbs are semantically interchangeable. This type of event-based similarity is not encoded in WordNet.

In cases where there is no proposition extracted for hypothesis and/or text sentences, the whole hypothesis and/or text sentences are taken to the step of entailment checking after their terms are lemmatized. In such cases, we use the *Levenstein* edit Distance (LD) between the hypothesis and the text. We use a shallow procedure where the LD distance takes characters as arguments. If the LD distance between a hypothesis and a text sentence is lower than a predefined threshold, then we infer that the text entails the hypothesis.

4 Experiments

4.1 Data

We have run our RTE system on three datasets provided by the *Text Analysis Conference* (TAC)[3] for the RTE track:

- TAC-RTE 2008 *test* dataset (rte4 - test), that includes 1000 pairs of hypotheses and texts.
- TAC-RTE 2009 main task *development* dataset (rte5 - dev.), that includes 600 pairs of hypotheses and texts.
- TAC-RTE 2009 main task *test* dataset (rte5 - test), that includes 600 pairs of hypotheses and texts.

4.2 Results

We have carried out experiments with our *baseline* RTE system where:

- The verbs are extended using FrameNet,
- The noun phrases are extended using WordNet,
- The WordNet distance threshold for finding hypernyms is equal to 1,
- The LD distance, in cases where proposition extraction fails, is equal to 3, and
- The term coverage procedure considers all terms in hypotheses (propositions) to have corresponding terms in texts (propositions).

In the TAC-RTE 2008 dataset, there are four categories of hypothesis-text pairs for Question Answering (QA), Information Extraction (IE), and Information Retrieval (IR), and Summarization (SUM)

[3] http://www.nist.gov/tac/

tasks. In the TAC-RTE 2009 datasets, however, there are only pairs for QA, IE, and IR tasks. We report the *accuracy* and the *recall* of our RTE system for these categories and the two classes *Entailment* and *No Entailment* in Table 2 and Table 3. For the RTE5 test dataset, we still do not have access to the answer set; therefore, recall cannot be measured at this stage.

Table 2: Accuracy of our baseline RTE runs on the RTE4 and RTE5 datasets – Avg. is a macro average

Dataset	Accuracy				
	QA	IE	IR	SUM	Avg.
rte4 - test	0.480	0.500	0.506	0.490	0.496
rte5 - dev.	0.480	0.470	0.520	N/A	0.490
rte5 - test	0.485	0.505	0.510	N/A	0.500

Table 3: Detailed analysis of our baseline RTE runs on the RTE4 and RTE5 datasets

Dataset	Correctly classified		Recall	
	ent.	No ent.	ent.	No ent.
rte4 - test	70	426	0.140	0.852
rte5 - dev.	25	269	0.083	0.896
rte5 - test	N/A	N/A	N/A	N/A

4.3 Discussion

As shown in Table 2, an average accuracy of 0.500 on the RTE5 test dataset is our best achievement so far where in our previous runs our baseline RTE system achieves an average accuracy of 0.496 and 0.490 for the RTE4 test and RTE5 development datasets.

A more detailed analysis of these results in Table 3 shows that our RTE system has not been very successful in recognizing correct entailment relationships. On the RTE4 test dataset, the entailment recall of 0.140 for 70 correctly classified items (out of 500 pairs) and on the RTE5 development dataset, the entailment recall of 0.083 for only 25 correctly classified items (out of 300 pairs) do not show high effectiveness in entailment recognition. Although with the accuracy measures obtained for the RTE5 test dataset we expect to see comparable classification performance and recall measures for the RTE5 test dataset, we do not have access to the gold standard test set and cannot report on these items for this dataset.

The overall statistics of the TAC-RTE 2009 systems shows the high, median, and low 2-way classification accuracies of 0.7350, 0.6117, and 0.5000 respectively. The overall performance of our RTE system does not reach high levels of accuracy, compared with the TAC-RTE 2009 statistics. We have conducted a failure analysis process to understand the underlying difficulty of the system.

4.4 System failure analysis

We have carried out an error analysis process of our baseline RTE system on the RTE4 test and the RTE5 development and test datasets with particular attention to syntactic parsing leading to proposition extraction. Table 4 summarizes the result of this analysis where *hypo* stands for hypothesis and *both* refers to the intersection of the sets of hypotheses and texts. The major barrier that interferes with our RTE system's performance seems to be the syntactic parsing stage where for the RTE4 test dataset, there are 131+320-57=394 hypotheses and texts for which no parses are returned by LGP. Therefore, the system has access to the parse of only ∼60% of the dataset to extract propositions. For the RTE5 development dataset this ratio is ∼80% of the dataset.

From another viewpoint, for the RTE4 test dataset there are 453+574-261=766 hypotheses and texts together where no propositions can be extracted for either the hypothesis or the text sentences. As a result, the semantic expansion and entailment checking procedures have access to proposition-level information for ∼23% of the pairs in the RTE4 test dataset. For the RTE5 development dataset, this ratio is ∼29% of the pairs.

Table 4: Error analysis of our baseline RTE runs on the RTE4 test and the RTE5 development datasets

Dataset	No parse			No prop.		
	hypo	text	both	hypo	text	both
rte4 - test	131	320	57	453	574	261
rte5 - dev.	58	60	2	352	192	119

We believe that, to improve the effectiveness of our lexical (term-based) RTE system, there is a need for further elaboration in two aspects:

- *Syntactic parsing*, using a more capable parser that is less sensitive to the grammati-

cal/structural flaws in texts and can more effectively handle long sentences, and

- *Proposition extraction*, by extracting/learning and utilizing a greater number of rules to extract propositions from parsed sentences.

5 Conclusion

A lexical Recognizing Textual Entailment (RTE) system participated in the Text Analysis Conference (TAC) 2009 has been introduced in this paper. This 2-way RTE system utilizes a syntactic approach prior to the term-based analysis of the hypotheses and texts in identification of entailment relationships.

The results of our RTE system on three datasets of the TAC-RTE tracks have been reported and shown moderate performances for our system. We have carried out a failure analysis of this RTE system to understand the underlying difficulties that interfere with the system performances. This has shown that the syntactic analysis of the hypotheses and texts, where sentences are parsed and propositions are extracted, is the main challenge that our system faces at this stage.

References

Elena Akhmatova and Diego Molla. 2006. Recognizing textual entailment via atomic propositions. In *Proceedings of the Machine Learning Challenges Workshop (MLCW)*, 385–403. Southampton, UK.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING*, 86–90. Universite de Montreal, Montreal, Quebec, Canada.

W.E. Bosma and C. Callison-Burch. 2006. Paraphrase substitution for recognizing textual entailment. In *Working Notes of CLEF 2006*, 1–8. Alicante, Spain.

Peter Clark and Phil Harrison. 2008. Recognizing textual entailment with logic inference. In *Proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment*. Gaithersburg, Maryland, USA.

M. Kouylekov and B. Magnini. 2005. Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the First PASCAL Challenges Workshop on Recognizing Textual Entailment*, 17–20. Southampton, UK.

D. Lin and P. Pantel. 2001. DIRT - Discovery of inference rules from text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 323–328. San Francisco, California, USA.

George A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

M. T. Pazienza, M. Pennacchiotti, and F. M. Zanzotto. 2005. Textual entailment as syntactic graph distance: A rule based and a SVM based approach. In *Proceedings of the First PASCAL Challenges Workshop on Recognizing Textual Entailment*, 25–28. Southampton, UK.

Alvaro Rodrigo, Anselmo Penas, and Felisa Verdejo. 2008. Towards an entity-based recognition of textual entailment. In *Proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment*. Gaithersburg, Maryland, USA.

R. de Salvo Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons. 2005. Textual entailment recognition based on dependency analysis and WordNet. In *Proceedings of the First PASCAL Challenges Workshop on Recognizing Textual Entailment*, 29–32. Southampton, UK.

Daniel Sleator and Davy Temperley. 1993. Parsing English with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technologies*.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the Conference on New Methods in Language Processing*. Manchester, UK.

Marta Tatu and Dan Moldovan. 2005. A semantic approach to recognizing textual entailment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, 371–378. Vancouver, British Columbia, Canada.

Mehmet Ali Yatbaz. 2008. RTE4: Normalized dependency tree alignment using unsupervised n-gram word similarity score. In *Proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment*. Gaithersburg, Maryland, USA.