STUDENT RESEARCH WORKSHOP

held in conjunction with

The International Conference RANLP - 2009

PROCEEDINGS

Edited by

Irina Temnikova, Ivelina Nikolova
and Natalia Konstantinova

Borovets, Bulgaria

14-15 September 2009

Student Research
Workshop

PROCEEDINGS

Borovets, Bulgaria
14-15 September 2009

ISBN 978-954-452-014-4

Designed and Printed by INCOMA Ltd.
Shoumen, Bulgaria

ORGANISING COMMITTEE

The Student Research Workshop associated with RANLP-09 is organised by:

Irina Temnikova, Research Group in Computational Linguistics
Research Institute in Information and Language Processing (RIILP)
University of Wolverhampton, United Kingdom

Ivelina Nikolova, Linguistic Modelling Department, Institute for Parallel Processing,
Bulgarian Academy of Sciences, Bulgaria

Natalia Konstantinova, Research Group in Computational Linguistics
Research Institute in Information and Language Processing (RIILP)
University of Wolverhampton, United Kingdom

PROGRAMME COMMITTEE

Itziar Aldabe (University of the Basque Country, Spain)
Galia Angelova (Bulgarian Academy of Sciences, Bulgaria)
Chris Biemann (Powerset / Microsoft, San Francisco, USA)
Iustin Dornescu (University of Wolverhampton, UK)
Le An Ha (University of Wolverhampton, UK)
Laura Hasler (University of Wolverhampton, UK)
Diana Inkpen (University of Ottawa, Canada)
Natalia Konstantinova (University of Wolverhampton, UK)
Sandra Kübler (Indiana University, USA)
Elina Lagoudaki (Imperial College, UK)
Wolfgang Maier (University of Tübingen, Germany)
Dalila Mekhaldi (University of Wolverhampton, UK)
Preslav Nakov (National University of Singapore, Singapore)
Ivelina Nikolova (Bulgarian Academy of Sciences, Bulgaria)
Constantin Orăsan (University of Wolverhampton, UK)
Petya Osenova (Bulgarian Academy of Sciences, Bulgaria)
Elena Paskaleva (Bulgarian Academy of Sciences, Bulgaria)
Viktor Pekar (Oxford University Press, UK)
Jelena Prokić (University of Groningen, The Netherlands)
Georg Rehm (vionto GmbH, Berlin, Germany)
Lucia Specia (University of Wolverhampton, UK)
Irina Temnikova (University of Wolverhampton, UK)
Cristina Toledo (University of Málaga, Spain)
Pinar Wennerberg (Siemens AG, Germany)

PREFACE

The Recent Advances in Natural Language Processing (RANLP) conference, already in its seventh year and ranked among the most influential NLP conferences, has always been a meeting venue for scientists coming from all over the world. In the last few years there has been increasing interest from students and young researchers from the NLP field. For this reason, for the first time in RANLP history, a competitive Student Workshop featuring peer reviewing and printed proceedings is held in conjunction with the International conference RANLP 2009.

The aim of the workshop is to provide an excellent opportunity for students at all levels (Bachelor, Master, and PhD) to present their work in progress or completed projects to an international research audience and receive feedback from senior researchers. We have received 29 high quality submissions, among which only 3 papers have been accepted as regular oral papers, and 13 as posters. Each submission has been reviewed by at least 2 reviewers, who are experts in their field, in order to supply detailed and helpful comments. The papers' topics cover a broad selection of research areas, such as:

- Anaphora Resolution;
- Information Extraction;
- Information Retrieval;
- Machine Translation;
- Multiple-Choice Questions Generation;
- Opinion Mining;
- Parsing;
- Part-of-Speech Tagging;
- Temporal Processing;
- Text Segmentation;
- Text Summarization;
- Textual Entailment;
- Word Sense Disambiguation.

We are also glad to admit that our authors comprise a very international group with students coming from: Argentina, Bulgaria, China, France, India, Iran, Italy, Portugal, Romania, Spain, Tunisia, Turkey, United Kingdom and United States.

We would like to thank the authors for submitting their papers and the members of the Programme Committee for their efforts to provide exhaustive reviews.

We are especially grateful to the RANLP Chairs Prof. Galia Angelova and Prof. Ruslan Mitkov, and to Dr. Constantin Orăsan for their indispensable support and encouragement during the Workshop organisation.

We hope the participants will receive invaluable feedback about their work. Enjoy the Workshop!

Irina Temnikova, Ivelina Nikolova and Natalia Konstantinova
Organisers of the Student Workshop, held in conjunction with
The International Conference RANLP-09

Table of Contents

Workshop Programme

14 September 2009

Poster presentations - 17:15–18:30

Normalized Accessor Variety Combined with Conditional Random Fields in Chinese Word Segmentation
Saike HE, Taozheng ZHANG, Xue BAI, Xiaojie WANG and Yuan DONG

LOGICON: A System for Extracting Semantic Structure using Partial Parsing
Kais DUKES

An Evaluation of Output Quality of Machine Translation Program
Mitra SHAHABI

Ambiguous Arabic Words Disambiguation: The Results
Laroussi MERHBEN, Anis ZOUAGHI and Mounir ZRIGUI

Hierarchical Discourse Parsing Based on Similarity Metrics
Ravikiran VADLAPUDI, Poornima MALEPATI and Suman YELATI

Effect of Minimal Semantics on Dependency Parsing
Bharat Ram AMBATI, Pujitha GADE, Chaitanya GSK and Samar HUSAIN

A Study of Machine Learning Algorithms for Recognizing Textual Entailment
Julio Javier CASTILLO

Exploring Context Variation and Lexicon Coverage in Projection-Based Approach for Term Translation
Raphaël RUBINO

ZAC.PB: An Annotated Corpus for Zero Anaphora Resolution in Portuguese
Simone PEREIRA

A Rule-Based Approach to the Identification of Spanish Zero Pronouns
Luz RELLO and Iustina ILISEI

Context Driven XML Retrieval
Aneliya TINCHEVA

Framework for using a Natural Language Approach to Object Identification
Mosa Emhamed ELBENDAK

Improving the Output from Software that Generates Multiple Choice Question (MCQ) Test Items Automatically using Controlled Rhetorical Structure Theory
Robert Michael FOSTER

15 September 2009

Oral Presentations

18:15–18:45 *Event Ordering. Temporal Annotation on Top of the BulTreeBank*
Laska LASKOVA

17:15–17:45 *A Two-stage Bootstrapping Algorithm for Relation Extraction*
Ang SUN

17:45–18:15 *Mink: An Incremental Data-Driven Dependency Parser with Integrated Conversion to Semantics*
Rachael CANTRELL

Effect of Minimal Semantics on Dependency Parsing

Bharat Ram Ambati
LTRC
IIIT-Hyderabad
India
ambati@students.iiit.ac.in

Pujitha Gade
LTRC
IIIT-Hyderabad
India
pujitha@students.iiit.ac.in

Chaitanya GSK
LTRC
IIIT-Hyderabad
India
chaitanya_gsk@students.iiit.ac.in

Samar Husain
LTRC
IIIT-Hyderabad
India
samar@research.iiit.ac.in

Abstract

In many languages general syntactic cues are insufficient to disambiguate crucial relations in the task of Parsing. In such cases semantics is necessary. In this paper we show the effect of minimal semantics on parsing. We did experiments on Hindi, a morphologically rich free word order language to show this effect. We conducted experiments with the two data-driven parsers MSTPaser and MaltParser. We did all the experiments on a part of Hyderabad Dependency Treebank. With the introduction of minimal semantics we achieved an increase of 1.65% and 2.01% in labeled attachment score and labeled accuracy respectively over state-of-the-art data driven dependency parser.

Keywords
Minimal semantics, dependency parsing, free word order language, Malt parser, MST parser.

1. Introduction

Parsing morphologically rich free word order language like Hindi[1] is a challenging task. For such languages dependency based framework suits better than the constituency based one [10][19][15][2]. Data driven dependency parsers has achieved considerable success due to the availability of annotated corpora in recent years. In spite of availability of annotated treebanks, state-of-the-art parsers for these languages have not reached the performance obtained for English [16]. Small size of the treebanks, non-projectivity, complex linguistic phenomenon, long distance dependencies and lack of explicit cues are most frequently stated reasons for low performance [16][17][3].

Previously, [5] used semantic features and showed an improvement in error reduction in the LAS[2] up to 5.8% with a dependency parser trained on the WSJ Penn Treebank sections 2-21 [12]. For Hindi, [3] showed that two semantic features namely, animate and in-animate, can reduce the subject-object confusion to a large extent.

In many languages such as Hindi, general syntactic cues are sometimes insufficient to disambiguate crucial relations in the task of parsing. For such cases semantics is necessary. In this paper we try to investigate the role of minimal semantics in parsing and try to ascertain its contribution to parsing accuracy. All our experiments are on Hindi, a morphologically rich free word order language. We conducted experiments with the two data driven parsers MSTParser and MaltParser. Part of Hyderabad Dependency Treebank [1] has been used as the experimental data. With the introduction of minimal semantics there was an increase of 1.65% and 2.01% in labeled attachment score and labeled accuracy respectively over state-of-the-art data driven Hindi dependency parser [3].

The paper is arranged as follows, Section 2 gives a brief overview of the necessity of semantics. Section 3 briefly describes the semantic tagset. Section 4 presents the experiments. In Section 5 we discuss our observations. We conclude our paper with future work in Section 6.

2. Why Semantics?

To elegantly describe the varied phenomena of structure of Hindi, it is analyzed in Paninian framework [2][1]. This framework employs syntactico-semantic relations named *karakas*. karakas are syntactico-semantic in nature. Consequently, both Hindi dependency annotation [1] and dependency parsing follow this framework [3][4]. In this scheme, vibhakti[3] and TAM[4] are the crucial markers which help in identifying the correct dependency label [3]. Sometimes though, this information is unavailable to help

[1] Hindi is a verb final language with free word order and a rich case marking system. It is one of the official languages of India, and is spoken by ~800 million people.

[2] Labeled attachment score

[3] A generic term for prepositions, post-positions, suffixes.

[4] Tense, aspect and modality.

disambiguate conflicting relations. In Hindi, this might happen, for example, when the lexical items have no postpositions. Take the following example:

(1) *raama* *seba* *khaataa* *hai*
‘Ram’ ‘apple’ ‘eat’ ‘is’
‘Ram eats apple’

In (1), both ‘*raama*’ and ‘*seba*’ have ø post-position and therefore there are no explicit syntactic cues that tell us that ‘*raama*’ is eating ‘*seba*’. Compare this with (2), where ‘*seba*’ is followed by a postposition that can help us identify the object/theme of the event ‘eat’,

(2) *raama* *seba* *ko* *khaataa* *hai*
‘Ram’ ‘apple’ ‘ACC’ ‘eat’ ‘is’
‘Ram eats apple’

It should also be noted that in (1) the agreement information doesn’t help much as both the elements are masculine. Neither will word order help, as Hindi is a free word order language. In such cases where syntactic information fails, semantic features assist in disambiguating the labels, thus aiding parsing. In (1), for example, the information that ‘*raama*’ is a human or that ‘*seba*’ is an inanimate being will prove to be crucial. In fact, in (1), correct parsing is only possible if this semantic information is available. All the semantic features don't contribute in identifying the dependency relations. Similarly, all the dependency relations are not benefited by semantic features. So an optimal set of semantic features should be used based on their positive contribution in dependency parsing. This paper is the first attempt in this direction.

3. Semantic Tags

The semantic tagset that we use in our experiments have been selected based on their closeness to the dependency labels that the parser is supposed to identify. In the Paninian framework all potential participants can participate in an action in six possible ways [2]. We have consequently kept this in mind while formulating the tagset. We call this set ‘minimal’ as the semantic labels broadly correspond to the semantic type of some of the core arguments. In this section we describe the semantic tags used for annotating the data. We also describe how these tags can help in disambiguating conflicts between some dependency labels.

3.1 Human

This tag is used to mark all the nouns which represent humans.
Eg: *mai* ‘I’, *lekhaka* ‘Author’

3.2 Non human

This tag is used to mark all the nouns which are animate but not human.
Eg: *kuttaa* ‘Dog’

3.3 Inanimate

This tag is used to mark all the inanimate nouns.
Eg: *kahaanii* ’Story’, *seba* ’apple’

The above three tags help in reducing k1-k2[5] ambiguity. We saw the significance of such tags clearly in example (1) previously. Below we repeat (1) and show how k1-k2 disambiguation is done by semantic features.

(3) *raama* *seba* *khaataa* *hai*
(sem-h) **(sem-in)**
‘Ram’ ‘apple’ ‘eat’ ‘is’
‘Ram eats apple’

In the above example both the nouns ‘*raama*’ and ‘*seba*’ have same vibhakti (ø). Position cannot help in disambiguation due to free word order property of Hindi. Also, agreement info. doesn’t help either. Semantic tags of ‘*raama*’ and ‘*seba*’ are human and inanimate respectively. With the introduction of semantics, ‘*raama*’ that is marked as human can be identified as a k1 and ‘*seba*’ as k2.

These semantic tags also help in disambiguating pof (part-of relation) relation from k1or k2. The relation between the nominal particle and its light verb is identified by a pof.

3.4 Time

All the nouns referring to time are marked with this tag. This helps in dependency labeling of the nouns with the label k7t. It indicates the time of the action.

Eg: *aaja* ‘Today’, *saala* ‘Year’

3.5 Place

All the nouns referring to place are marked with this tag. This helps in dependency labeling of the nouns with the label k7p. It is used to indicate the place where the action took place.

Eg: *skuula* ‘School’, *kheta* ‘Field’

[5] k1 (*karta*) and k2 (*karma*) are syntactico-semantic labels which have some properties of both grammatical roles and thematic roles. k1 for example, behaves similar to subject and agent. k2 behaves similar to object and patient. For the complete tagset description, see: http://ltrc.iiit.ac.in/MachineTrans/publications/technicalReports/tr032/treebank.pdf

As nouns are explicitly marked with time and place semantic labels, identifying the dependency relations k7p and k7t becomes very easy. The features time and place not only reduce ambiguity between k7p and k7t but also reduce the ambiguity of these two labels with other labels like k1, k2 etc.

(4) *puraaNe samaya meM jMgala meM eka shera*
‘ancient’ ‘time’ ‘ in’ ‘jungle’ ‘in’ ‘a’ ‘lion’
thaa
‘was’
(sem-t) **(sem-p)**
‘In ancient times there was a lion in the jungle.’

In the above example *samaya* and *jMgala*, both the noun have ‘*meM*’ vibhakti. Semantic tags of ‘*samaya*’ and ‘*jMgala*’ are time and place respectively. Prediction of dependency labels k7p and k7t from these tags becomes straight forward.

3.6 Abstract

All the abstract nouns are annotated with the abstract tag. Eg: *kaama* ‘Work’, *racanaa* ‘Literary work’, *vicaara* ‘Thought’

3.7 Rest

The rest of the nouns which do not fall into any of the above categories are marked with the tag ‘rest’.

4. Experimental Setup

4.1 Parsers

We performed experiments with two data-driven parsers MaltParser[6] [16] and MSTParser[7] [14].

MaltParser is a transition based parser. It is a Shift/Reduce parser. It uses graph transformation to handle non-projective trees. MST has an implementation of Chu-Lui-Edmonds MST algorithm [6][8]. It uses online large margin learning as the learning algorithm [13]. Both the parsers provide an option for using different combinations of features.

4.2 Data

For our experiments we extracted 1221 sentences from HyDT [1], the dependency treebank for Hindi. Average length of these sentences is 17.83 words/sentence and 9.04 chunks/sentence. In HyDT, chunk heads appear as nodes. A chunk is a set of adjacent words which are in dependency relation with each other, and are connected to the rest of the words by a single incoming arc to the chunk. All noun chunks are manually annotated with one of the 7 semantic categories (see Section 3). We divided this data in to training, development and testing data each containing 850, 200 and 171 sentences respectively. We used experiment F3 for FEATS as this feature gives best results [3]. F3 uses TAM class for verb chunks, or vibhakti for noun chunks. We provide this feature in the FEATS column of the CoNLL format.

4.3 Experiments and Results

Both Malt and MST parsers were used during the experiments. We tried out different parser settings and applied the best one on test set. For MSTParser, non-projective algorithm, order=2 and training-k=5 gave best results. For feature model, we used conjoined feature set of [3]. For Malt Parser, arc-eager gave better performance over other algorithms. For feature model we tried out different combinations of best feature settings of the same parser on different languages in CoNLL-2007 shared task [9] and applied the best feature model on the test data. In FEATS column of CoNLL format in addition to F3, we appended manually annotated semantic categories. We name this feature as F5. Hence F5 is F3+semantic features. Results of both F3 and F5 can be seen in Table1. We evaluated our experiments based on unlabeled attachment score (UAS), labeled attachment score (LAS) and label accuracy (L).

Example 2:
puraaNe samaya meM jMgala meM eka shera
‘ancient’ ‘time’ ‘ in’ ‘jungle’ ‘in’ ‘a’ ‘lion’
(sem-t)(sem-p)
rehtha thaa
‘live’ ‘was’
‘In ancient times there lived a lion in the jungle.’

In the above example samaya and jMgala, both the noun chunks have ‘meM’ vibhakti. Semantic tags of ‘samaya’ and ‘jMgala’ are time and place respectively. Prediction of dependency labels k7p and k7t from place and time semantic labels is pretty easy.

Table 1. Results of MST and Malt Parsers

	MST			**Malt**		
	UAS	**LAS**	**L**	**UAS**	**LAS**	**L**

[6] Malt Version 1.2 (http://maltparser.org/download.html)

[7] MST Version 0.4b (http://sourceforge.net/projects/mstparser/)

F3	86.77	65.28	69.16	88.57	69.81	72.68
F5	**87.13**	**69.45**	**73.04**	**88.21**	**71.46**	**74.69**

5. Discussion

With the introduction of semantic features there is a significant improvement in the performance of both the parsers. As expected, adding semantic features for nouns helps with label identification more than head identification. This is clearly shown by the improvements in LAS vs. UAS. For MST, there is an increase of 0.36% in UAS, 4.17% in LAS and 3.88% in L. Similarly, in case of Malt, there is an increase of 1.65% in LAS and 2.01% in L. These results clearly show that minimal semantics can help in improving the parsing accuracy. More importantly, this information in many instances cannot be done away with to get the correct parse.

Table 2 and Table 3 show the precision, recall and f-measures of the dependency labels k1, k2, pof, k7p and k7t for both the parsers. The results show that the seven semantic labels considered are indeed crucial to reduce ambiguities among these five labels.

Table 2. Tag-wise LAS for MSTParser

		MST Parser		
		Precision	**Recall**	**Fβ=1**
k1	**F3**	71.59	67.15	69.29
	F5	**77.82**	**71.68**	**74.62**
k2	**F3**	53.57	50.97	52.24
	F5	**57.65**	**61.75**	**59.63**
pof	**F3**	74.51	43.18	54.67
	F5	**88.24**	**56.96**	**69.23**
k7p	**F3**	50.00	64.18	56.21
	F5	**67.44**	**68.24**	**67.84**
k7t	**F3**	74.55	89.13	81.19
	F5	**83.64**	**74.19**	**78.63**

Table 3. Tag-wise LAS for Malt Parser

		Malt Parser		
		Precision	**Recall**	**Fβ=1**
k1	**F3**	77.22	77.82	77.52
	F5	**77.86**	**82.10**	**79.92**
k2	**F3**	59.91	66.33	62.96
	F5	**63.05**	**65.31**	**64.16**
pof	**F3**	44.71	74.51	55.89
	F5	**51.32**	**76.47**	**61.42**
k7p	**F3**	68.18	52.33	59.21
	F5	**72.15**	**66.28**	**69.09**
k7t	**F3**	80.85	69.09	74.51
	F5	**73.21**	**74.55**	**73.87**

6. Conclusion and Future Work

This paper clearly shows that minimal semantics helps in boosting the parsing accuracy. Instead of manually annotated semantic labels, we have to experiment with automatically extracted semantic labels. One way to get these labels is from the first sense of the words in Hindi WordNet [11]. Other method is to use an automatic semantic labeler [18]. We can experiment with iterative learning between Dependency Parsing and Semantic Labeling [7].

7. Acknowledgements

We would like to express our gratitude to Dr. Dipti Misra Sharma and Prof. Rajeev Sangal for their guidance and support. We would also like to thank Itisree Jena and Sushant Kishore for manually annotating the Treebank with semantic features.

References

[1] R. Begum, S. Husain, A. Dhwaj, D. Sharma, L. Bai, and R. Sangal. 2008. Dependency annotation scheme for Indian languages. In Proceedings of IJCNLP-2008.

[2] A. Bharati, V. Chaitanya and R. Sangal. 1995. Natural Language Processing: A Paninian Perspective, Prentice-Hall of India, New Delhi.

[3] A. Bharati, S. Husain, B. Ambati, S. Jain, D. Sharma, and R. Sangal. 2008a. Two semantic features make all the difference in parsing accuracy. In Proceedings of ICON-08.

[4] A. Bharati, S. Husain, D. M. Sharma, and R. Sangal. 2008b. A Two-Stage Constraint Based Dependency Parser for Free Word Order Languages. In Proceedings of the COLIPS IALP, Chiang Mai, Thailand.

[5] M. Ciaramita and G. Attardi. 2007. Dependency Parsing with Second-Order Feature Maps and Annotated Semantic Information. In *Proceedings of IWPT 2007.*

[6] Y.J. Chu and T.H. Liu. 1965. On the shortest arbores-cence of a directed graph. Science Sinica, 14:1396–1400.

[7] Q. Dai, E. Chen, and L. Shi. 2009. An iterative approach for joint dependency parsing and se-mantic role labeling. In Proceedings of the 13th Con-ference on Computational Natural Language Learning*(CoNLL-2009), June 4-5,* Boulder, Colorado, USA.June 4-5.

[8] J. Edmonds. 1967. Optimum branchings. Journal of Research of the National Bureau of Standards, 71B:233–240.

[9] J. Hall, J. Nilsson, J. Nivre, G. Eryigit, B. Megyesi, M. Nilsson and M. Saers. 2007. Single Malt or Blended? A Study in Multilingual Parser Optimization. In Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL.

[10] J. Hudson, 1984. 'Why English should be taught as a second language in Aboriginal schools in the Kimberleys', pp.99-106 in *Wikaru*, Vol.12.

[11] S. Jha, D. Narayan, P. Pande and P. A. Bhattacharyya. 2001. WordNet for Hindi. International Workshop on Lexical Resources in Natural Language Processing, Hyderabad, India, January 2001.

[12] M. Marcus, B. Santorini, and M. Marcinkiewicz. (1993). Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19:313-330.

[13] R. McDonald, K. Crammer, and F. Pereira. 2005b. Online large-margin training of dependency parsers. In Proceedings of ACL. pp. 91–98.

[14] R. McDonald, F. Pereira, K. Ribarov, and J. Hajic. 2005a. Non-projective dependency parsing using spanning tree algorithms. Proceedings of HLT/EMNLP, pp. 523–530.

[15] I. A. Mel'Cuk. 1988. Dependency Syntax: Theory and Practice, State University Press of New York.

[16] J. Nivre, J. Hall, S. Kubler, R. McDonald, J. Nilsson, S. Riedel and D. Yuret. 2007b. The CoNLL 2007 Shared Task on Dependency Parsing. In Proceedings of EMNLP/CoNLL-2007.

[17] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S.Marinov and E Marsi. 2007a. MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering, 13(2), 95-135.

[18] S. Reddy, A. Inumella, R. Sangal and S. Paul. 2009. All Words Unsupervised Semantic Category Labeling for Hindi. To be appeared in Recent

[19] S. M. Shieber. 1985. Evidence against the context-freeness of natural language. In Linguistics and Philosophy, p. 8, 334–343.

Mink: An Incremental Data-Driven Dependency Parser with Integrated Conversion to Semantics

Rachael Cantrell
Indiana University
Bloomington, Indiana
rcantrel@indiana.edu

Abstract

While there are several data-driven dependency parsers, there is still a gap with regards to incrementality. However, as shown in Brick and Scheutz [3], incremental processing is necessary in human-robot interaction. As is shown in Nivre et al. [12], dependency parsing is well-suited for mostly incremental processing. However, there is as of yet no dependency parser that combines syntax and semantics by including traditional dependency parsing, CCG tagging, and lambda-logical structures in one fast, accurate application suitable for embodied natural language processing.

This paper addresses that gap by introducing Mink, an incremental data-driven dependency parser with integrated conversion to semantics. We show that Mink is comparable to similar but non-incremental parsers, and that it succeeds at performing some semantic analysis.

Keywords

incremental parsing, dependency parsing, CCG, semantics conversion

1 Introduction

Dependency parsing has gained in popularity in the last few years. However, there is still a gap with regards to incrementality. As is shown in Nivre et al. [12], dependency parsing is well-suited for mostly incremental processing. However, there is as of yet no dependency parser that combines syntax and semantics by including traditional dependency parsing, CCG tagging, and lambda-logical structures in one fast, accurate application suitable for embodied natural language processing.

Our motivation for creating Mink was to meet the need for incremental processing in robots that must interact with humans using natural language. Most natural language processing (NLP) is performed on written texts. However, in embodied NLP systems such as robots, the system needs to understand incoming speech from humans. As discussed by Brick and Scheutz [3], humans do not wait for a full sentence in order to begin processing the sentence, parsing it, and resolving references. We immediately begin to process whatever we can as soon as we can, which allows us to look at referents, or reach for things before we know where we'll be told to put them, etc. Because humans have this capability, we become impatient when dealing with conversation partners that are slower. Mink takes us one step closer toward achieving a level of incremental processing in robots that will allow humans and robots to communicate easily and naturally.

Because we are working to help our robots "understand" verbal input, we are not interested in the parses themselves, but rather in what they can tell us semantically. Thus we have integrated semantic output into the system itself. Mink outputs three different types of information: dependency arcs, combinatorial categorial grammar (CCG) tags, and, when possible, lambda-logical semantic conversions.

In this paper, we evaluate Mink based on accuracy and speed. We evaluate the CCG tagging and semantic conversion capabilities together, by using them to attempt to correctly classify utterances into either sentence, question, or command. We chose these simple types because they are discourse types that are, in general, distinguishable from the syntax of the sentence.

In section 2, we discuss currently available dependency parsers, CCG taggers and parsers, and relevant work on semantics, particularly the interface between syntax and semantics. In section 3, we discuss the details of our parser. In section 4, we evaluate our parser. Finally, in section 5, we discuss how we are currently able to use the parser and where we plan to go with it.

2 Related Work

There are three main bodies of knowledge from which this project draws: dependency parsing, CCG tagging and parsing, and semantic representation of parser output.

2.1 Dependency Parsing

One often-used data-driven dependency parser is MaltParser [13], a shift-reduce dependency parser that uses support vector machines to deduce, from training examples, what next action to take in parsing a sentence. Nivre [12] discusses the potential for incrementality in MaltParser. It is determined that the algorithm is the best suited for incremental parsing that has been developed to date; however an incremental version of MaltParser has not been developed by those researchers.

MSTParser (c.f. McDonald et al. [10]) approaches the problem of finding dependency trees as one of finding maximum spanning trees. While this method is shown to be successful, the fact that it searches the entire space of possible dependency trees renders it inefficient for our purposes.

Johannsen and Nugues [8] combine aspects of MaltParser and MSTParser to form a dependency parser that, rather than maximizing the probability of each individual action (like MaltParser), maximizes the probability over a complete sentence. While this gives good results, it is a step away from incrementality.

2.2 CCG Tagging and Parsing

An utterance that is tagged with CCG tags contains more information about the parse tree of the utterance than one tagged with simple POS tags. This is because each tag contains information about both the POS (the return type), and about the token's children (the argument types). In an utterance with just one possible parse, this is enough information to describe the correct parse tree. For example: $the_{NP/NP}$ $child_{NP}$ $ate_{S\backslash NP/NP}$ $a_{NP/NP}$ $snack_{NP}$.

There is just one way to create a tree from this information, so, even if this utterance came from a tagger rather than a parser, it is essentially parsed already. However, many utterances have ambiguity, which a tagger is not intended to resolve. For our purposes, in most cases, a tagger gives us the information we need; therefore we consider both taggers and parsers as being of equal utility to us.

There are two basic types of CCG taggers/parsers, statistical and rule-based. An example of a small rule-based system is found in [1]. It looks up the CCG tag(s) of each word and builds a parse based on combinatory rules. Another rule-based system is OpenCCG, a freely-available CCG parser written in Java. While it has been widely used in many projects[1], the nature of a rule-based system limits its robustness, particularly for processing spoken data, which often contains unknown words and disfluencies.

There are several statistic systems, which we would prefer for greater robustness. For example, Clark [5] uses a maximum-entropy based statistical tagger to select CCG tags based on context.

The problem with statistical systems, for our purposes, is that they are based on probabilities of full tags, rather than on the parts.

Because we wanted to be able to build the tags by combining individual parts rather than selecting from a set of full CCG tags, we decided to use a dependency parser to build the tags based on dependency arcs.

2.3 Semantic Representation

Che et al. [4] integrates syntax and semantics. They parse the sentence using MSTParser, then use a series of classifiers to identify predicates, classify them according to senses, and assign semantic roles to different elements in the sentence. However, they do not integrate the results into any sort of logical framework.

Bos et al. [2] discusses a method of converting output from a wide-coverage CCG parser into semantic representation using lambda calculus, from which our semantic conversion method draws heavily.

Lambda conversion has been shown to be semantically useful in robotic NLP systems, c.f. Gold and Scassellati [6].

3 An incremental architecture for dependency parsing and integrated semantics conversion

3.1 Parsing Algorithm

We used the Nivre algorithm, which was previously implemented in MaltParser [11], which itself is an adaptation of the shift-reduce parsing algorithm for constituency parsers.

The parser keeps track of what has already been input in a stack. Each time a token is input, the parser must decide whether to shift, reduce, create a left arc, or create a right arc. These actions are described in Table 1.

The parser continues processing until there is no more input and the stack is empty again, resulting in a connected, non-projective dependency graph. Currently, the parser needs to know when there is a pause or other possible indicator of a sentence boundary; however, an action is being developed that allows the parser to guess that it should terminate the current graph and begin work on a new one. This, rather than separating actual sentences, is intended to separate semantically meaningful fragments, since spoken data often consists of such fragments, rather than of complete sentences.

The main difference between our implementation and others is its incrementality. While the other implementations accept only completed text files and output the same, Mink accepts input from a stream, which in our case is the output of a speech recognizer, and outputs partial analyses as soon as they are available. As soon as it pops a token off the stack–i.e. as soon as it is clear the token will have no more dependents–it outputs the CCG tag and begins semantic conversion. It outputs each conversion as it makes it so that the module that mediates between semantic representation and action can immediately begin to process the semantics of the input to see what it can do with what it knows so far.

3.2 Machine Learning Algorithm and Features

To decide which actions to perform, we used a maximum-entropy based classifier, namely, the Logistic algorithm from the Weka Java-based machine learning library [15]. This classifier decides, for a pair of an input token and a token on the top of the stack, which action to perform.

We trained the classifier on the dependency graph version of the Penn Treebank, created with pennconverter [7].

[1] http://comp.ling.utexas.edu/wiki/doku.php/openccg/projects_using_openccg

Table 1: *Parser Actions*

Action	Description
Shift	Move the input token to the stack
Reduce	Pop the top token off the stack
Left-arc	Create a dependency arc pointing from the input to the token on the top of the stack; this is followed by a reduce action
Right-arc	Create a dependency arc pointing from the token on top of the stack to the input token; this is followed by a shift action

For features, we tested several different sets. First, we used the standard set of features used by MaltParser, since they have proven to be ideal for parsing English. These features are: the token on top of the stack (TOP), its part-of-speech tag and dependency type, the dependency types of its leftmost and rightmost dependents, the input token (NEXT), and its part-of-speech tag and dependency type, and the part-of-speech of the next + 1 input. Next, we eliminated the lookahead feature. Last, we eliminated labels and used only unlabeled arcs. See Table 2 for a complete listing of feature sets.

We evaluate each of these feature sets in section 4.

3.3 Building CCG tags

CCG tags give a bit more information than part-of-speech tags. Namely, they give both the return type, and the arguments. The return type is basically the part-of-speech tag, while the arguments are similar to phrasal constituents. Our method of constructing CCG tags from dependency arcs therefore involves two steps: determining the return type, and finding the arguments.

When a new token is input, an “empty” CCG tag is created. This includes the return type and list of arguments. Ultimately, it can accept 0 or more arguments. Currently, the return type is determined immediately upon creating the new tag in order to reduce the complexity of the computation. This is posed as a classification problem. The features used to make the classification are currently the token and WSJ POS tag that was already found by the tagger.

Whenever a dependency arc is discovered with a given token as the head, the dependent's return type is added as an argument to the head's CCG tag.

As an example, the phrase `the blue box` might be tagged and parsed as follows by a constituent parser using the WSJ tagset:

$(NP(the_{DT}\ blue_{JJ}\ box_{NN}))$.

However, with a CCG parser, it would be tagged and parsed

$(NP\ (the_{NP/NP}\ (NP\ blue_{NP/NP}\ (NP\ box_{NP}))))$.

In the future, we would like to evaluate the utility of determining the return type after finding the token's arguments in order to use the arguments as features; however, that has not yet been implemented.

The CCG representation of a parse tree is then used in the semantic conversion.

3.4 Semantic Conversions

The conversion to semantics relies on a semantic dictionary to translate from the parse trees to semantic representations. Each entry in the dictionary consists of a word, its possible CCG tags, and the lambda-logical expressions for each word/CCG tag combination.

The goal of the conversion is, in our case, to translate input sentences into appropriate responses to that input. For example, the question “Did you get it?” generates a *report(get+PST(I,box))* action, in which the robot reports the truth or falsity of the predicate *get+PST(I,box)*.

The conversion will fail if definitions retrieved from the dictionary do not form a connected parse. This is appropriate, since in this particular task, precision is more important than recall: if the robot does not understand an utterance, it can ask for clarification or request to have the utterance rephrased.

One problem of the conversion is that words can be ambiguous, according to human-level comprehension abilities. For example, take the sentence “there is another one in the corner to my right”. This could potentially be given two different readings: 1) the existential (“there exists another one in the corner to my right”), and 2) the locative (“another one is in the corner to my right”). This is a problem because the conversion must be deterministic: given an input, it must arrive at the same output every time. Because we currently choose definitions according to rule, this means that each word can have at most one definition for each word/valency combination. However, in the future, a statistical method of choosing the correct definition will be implemented.

4 Evaluation

There are three segments of the application to evaluate: the dependency parsing, the CCG tagging, and the semantic representation. The last two we evaluate simultaneously based on their success at identifying different types of utterances.

4.1 Data Set

In order to evaluate Mink on a domain relevant to our work, we use a spoken-data corpus of human-human dialogs [14], in which one person has to navigate a labyrinth and perform a task, guided by a second person outside the labyrinth. The two persons can only communicate via handheld devices. At present, the corpus comprises transcriptions of 12 dialogues.

4.2 Dependencies

We evaluate the accuracy of the dependency arcs by training both our parser and MaltParser on the same training material, then testing against identical test sets. We test MaltParser vs. each of the three feature sets from Table 2 and show that our parser is, in each

Table 2: *Feature Sets*

Features	Description	1	2	3
TOP.TOK	The token on top of the stack	y	y	y
TOP.POS	The part of speech of TOP	y	y	y
TOP.DEP	the dependency type of TOP	y	y	
TOP.LEFT	The dependency type of TOP's leftmost dependent	y	y	
TOP.RIGHT	The dependency type of TOP's rightmost dependent	y	y	
NEXT.TOK	The next input token	y	y	y
NEXT.POS	The part of speech of NEXT	y	y	y
NEXT.LEFT	The dependency type of NEXT's leftmost dependent	y	y	
LOOK.POS	The part-of-speech of the next plus one input	y		

case, comparable to MaltParser, as one would expect from a reimplementation of the algorithm.

In order to evaluate, we trained both parsers on the Penn treebank (PTB) [9], and tested them on 60 sentences from our corpus. The sentences are very much out of the PTB's domain, so this is purely for parser comparison. The sentences were randomly selected and tagged with Wall Street Journal (WSJ) POS tags using acopost[2], then hand-corrected in order to ensure correct POS tagging. The results can be seen in Table 3.

Table 3: *Parser Evaluation*

Metric	Malt	Set 1	Set 2	Set 3
Lbld attachmt	0.86	0.85	0.85	N/A
Unlbld attachmt	0.88	0.88	0.87	0.85
Parsing Time (s)	22.7	32.6	31.1	29.1

As can be seen, our results did not vary significantly from MaltParser's. Our results were just slightly lower, which we attribute to different machine-learning methods. Eliminating the lookahead feature had basically no effect on results, and eliminating all labeling lowered the unlabeled attachment score somewhat, though not significantly.

We also evaluate the speed of the system when running with either Mink or Malt. Since the system is used to respond to individual utterances rather than to process large texts, we evaluated this by parsing 5 sentences 10 times each and averaging the results. We show that adding incrementality so that processes can run in parallel, rather than in sequence, speeds up the entire process. The results of this evaluation are shown in Table 4.

Table 4: *Speed*

Utterance	Mink	Malt	Ratio
what is your goal	185	478.4	0.38
keep the lights on	144	534.5	0.26
cancel keep lights off	145.5	489.1	0.29
what are your orders	173.9	516	0.33
and try to report the locations of wounded people	304.8	525	0.58

4.3 CCG Tags and Semantics

For this particular task, since we are interested only in practical application, the best way of evaluating our tags is, rather than comparing them to those generated by other applications or to gold standards, seeing how successful they are at a practical task. We chose to use the parser to distinguish between different types of utterances, namely questions, commands, and statements. We chose not to use a more complex discourse scheme because we are investigating the interface between syntax, semantics, and discourse. More complex discourse schemes often make distinctions between discourse types that cannot be distinguished by syntax. That is outside the scope of the current project. Our last reason for choosing this particular task is because it is practically useful: we would like our robot to be able to respond to these types of utterances in different ways, by following commands, answering questions, and storing statements (facts that may be useful to it).

To this end, we use three different semantic dictionaries–a statement dictionary, a command dictionary, and a question dictionary–to semantically convert each statement in parallel. See Figures 5 and 6 for an example of a small dictionary, describing one question, one statement, and one command. The conversion itself fails if a given dictionary is not able to find a complete conversion for the statement. The classification fails if the utterance is correctly converted by multiple dictionaries, and thus it is not clear which type of utterance it is.

Table 5: *Sample utterances*

Type	Example
Question	$did_{S/S}\ you_{NP}\ get_{S\backslash NP/NP}\ it_{NP}$
Statement	$I_{NP}\ got_{S\backslash NP/NP}\ that_{NP/NP}\ one_{NP}$
Command	$get_{S/NP}\ that_{NP/NP}\ one_{NP}$

The obvious base for distinguishing between these types of utterances is in the verb subcategorization: Statements will, in general, have a subject to the left of the finite verb, while questions will, in general, have a subject to the right, and commands will not have an

[2] http://acopost.sourceforge.net/

Table 6: *Sample dictionary*

Word	CCG	Lambda expression
did	S/S	λ x. report(x)
get	S/NP,	λ x. get(you,x),
	S\NP/NP	λ x. λ y. get(x, y)
got	S\NP/NP	λ x. λ y. get+PST(x, y)
I	NP	I
it	NP	it
one	NP	one
that	NP/NP	λ x. find-reference(x)
you	NP	you

overt subject at all. There are, of course, exceptions to these generalities which will ultimately require a more complex method of distinguishing between them. In each case, only one semantic conversion should succeed, thus letting us know which type of utterance it is.

We evaluate 60 sentences–20 randomly chosen from each of three human-selected subsets: questions, statements, and commands– from our experimental corpus. There are four possibilities for each test example: it does not parse with any of the dictionaries; it parses with just one dictionary, and it is correctly classified; it parses with just one dictionary and it is incorrectly classified; it parses with multiple dictionaries and thus is left unclassified.

For determining which category a statement belonged to, we categorized only the explicit structure rather than the utterance's discoursal meaning. For example, "if you enter the closet you should see a box" can be seen an an implicit command to enter the closet and check for a box. However, it is explicitly a simple descriptive statement. Particularly in our case, where all utterance types are translated into some kind of response, the system needs to be able to translate only the *explicit* form of the statement into an action.

As can be seen from the results in Table 7, this method of classifying sentence types was quite successful. All utterances classified as either commands or questions were correctly classified; however, some questions and commands were classified as statements; and some statements were not classified at all.

Looking at the data, it is not surprising that questions were misclassified as statements; some question phrasing is very statement-like, and some other method will have to be deployed to identify these utterances. A few examples that were misclassified are: `we are not supposed to take the blue box though right just the blocks ?` and `so you went through the second room right ?` When humans hear such sentences, prosodic contours tell us how to interpret the sentences; but in the current system, we are not able to make use of such information.

The commands that were classified as statements were all similar to the following: `you leave it` and `and then you head back`. Again, though they appeared command-like to the human classifiers, it is obvious why they were misclassified as statements.

Table 7: *Semantics Evaluation*

Utterance type	precision	recall	f-score
Commands	1.00	0.90	0.94
Questions	1.00	0.70	0.82
Statements	0.69	0.90	0.78

5 Conclusion and Future Work

In this paper, we have described an incremental data-driven dependency parser that outputs graphs, CCG tags, and semantic representations of the input. We have shown that its accuracy is comparable to that of other data-driven dependency parsers, and that it is successful at creating useful CCG tags for practical semantic tasks.

As we move forward with this project, we will investigate more feature sets for the parser, in the hopes of finding the smallest possible set that continues to achieve high accuracy. Additionally, there are several aspects of our system that are rule-based, and we will investigate the feasibility of making every aspect statistical.

Finally, our system has largely been working with features that are extractable from the text of a dialogue. Clearly, in human-robot interaction, there is much that can be learned from other aspects of the interaction, in particular intonation. In the future, we will experiment with integrating prosodic features into the classification system.

References

[1] C. Baral, J. Dzifcak, and T. C. Son. Using answer set programming and lambda calculus to characterize natural language sentences with normatives and exceptions. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI)*, pages 818–823, Chicago, Illinois, 2008.

[2] J. Bos, S. Clark, M. Steedman, J. R. Curran, and J. Hockenmaier. Wide-coverage semantic representations from a ccg parser. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1240, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[3] T. Brick and M. Scheutz. Incremental natural language processing for hri. In *The Second ACM IEEE International Conference on Human-Robot Interaction*, pages 263–270, 2007.

[4] W. Che, Z. Li, Y. Hu, Y. Li, B. Qin, T. Liu, and S. Li. A cascaded syntactic and semantic dependency parsing system. In *CoNLL*, pages 238–242, Manchester, England, August 2008. Coling 2008 Organizing Committee.

[5] S. Clark. Supertagging for combinatory categorial grammar. In *In Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*, pages 19–24, 2002.

[6] K. Gold and B. Scassellati. A robot that uses existing vocabulary to infer non-visual word meanings from observation. In *The Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, 2007.

[7] R. Johansson and P. Nugues. Extended constituent-to-dependency conversion for English. In *NODALIDA 2007*, 2007.

[8] R. Johansson and P. Nugues. Incremental dependency parsing using online learning. In *The CoNLL Shared Task Session of EMNLP-CoNLL*, pages 1134–1138, 2007.

[9] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2), 1993.

[10] R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. Non-projective dependency parsing using spanning tree algorithms. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[11] J. Nivre. An efficient algorithm for projective dependency parsing. In *The 8th International Workshop on Parsing Technologies*, pages 149–160, 2003.

[12] J. Nivre. Incrementality in deterministic dependency parsing. *Incremental Parsing: Bringing Engineering and Cognition Together. Workshop at ACL-2004*, 2004.

[13] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryiğit, S. Kübler, S. Marinov, and E. Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[14] M. Scheutz and K. Eberhard. Towards a framework for integrated natural language processing architectures for social robots. In *The Fifth International Workshop on Natural Language Processing and Cognitive Science (NLPCS-2008)*, 2008.

[15] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann, San Francisco, 2005.

A Study of Machine Learning Algorithms for Recognizing Textual Entailment

Julio Javier Castillo

Faculty of Mathematic Astronomy and Physics - National University of Cordoba
Córdoba, Argentina
jotacastillo@gmail.com

Abstract

This paper presents a system that uses machine learning algorithms and a combination of data sets for the task of recognizing textual entailment. The chosen features quantify lexical, syntactic and semantic level by matching between texts and hypothesis sentences. Additionally, we created a filter which uses a set of heuristics based on Named Entities to detect cases where no entailment was found. We analyzed how the different sizes of data sets and classifiers could impact on the final overall performance of the systems.
We show that the system performs better than the baseline and the average of the systems from the RTE on both two and three way tasks.

We concluded that evaluating using the RTE3 test set, the model learned using MLP from the RTE3 alone outperforms other models that employed different ML algorithms and additional training data from the RTE1 and RTE 2.

Keywords
Textual entailment, machine learning, rte data sets.

1. Approach

The objective of the Recognizing Textual Entailment Challenge is determining whether the meaning of the Hypothesis (H) can be inferred from a text (T). Recently the RTE4 Challenge has changed to a 3-way task that consists in distinguish among entailment, contradiction and unknown when there is no information to accept or reject the hypothesis. However the traditional two-way distinction between entailment and non-entailment is still allowed.

In the past, RTEs Challenges machine learning algorithms were widely used for the task of recognizing textual entailment (Marneffe et al., Zanzotto et al.). Thus in this paper we tested the most common classifiers that have been used by other researchers in order to provide a common framework of evaluation of ML algorithms (fixing the features) and showing how the development data set could impact over them.

We generated a feature vector with the following components for both Text and Hypothesis:

- Levenshtein distance,
- Lexical level: a lexical distance based on Levenshtein,
- Semantic level: a semantic similarity measure Wordnet based,
- LCS (longest common substring) metric.

We chose only four features in order to learn the development sets. Larger feature sets do not necessarily lead to improving classification performance because it could increase the risk of overfitting the training data. In section 3 we provide a correlation analysis of these features.

The motivation of the input features:
Levenshtein distance is motivated by the good results obtained as a measure of similarity between two strings. Additionally, we proposed a lexical distance which is based on Levenshtein distance but working to sentence level.
We created a metric based on Wordnet to try to capture the semantic similarity between T and H to sentence level.
Longest common substring is selected because is easy to implement and provides a good measure for word overlap.
Furthermore, the system uses a NER filter that detects cases where no entailment relation is found. This filter applies heuristic rules over Named Entities found in the text and hypothesis.

The system produces feature vectors for all possible combinations of the available development data RTE1, RTE2 and RTE3. Weka (Witten and Frank, 2000) is used to train classifiers on these feature vectors. We experimented with the following five machine learning algorithms:

- Support Vector Machine (SVM),
- AdaBoost (AB),
- BayesNet (BN),
- Multilayer Perceptron (MLP),
- Decision Trees (DT).

The Decision Trees are interesting because we can see what features were selected from the top levels of the trees. SVM, Bayes Net and AdaBoost were selected because they are known for achieving high performances. MLP was used because has achieved high performance in others NLP tasks.

We experimented with various parameters (settings) for the machine learning algorithm, such like increasing the confidence factor in DT for more pruning of the trees, different configuration(layers and neurons) for the neural network, and different kernels for SVM. Thus, we tested classifiers used by other researchers in order to provide a common framework of evaluation.

For two-way classification task, we used the RTE1, RTE2, RTE3 development sets from Pascal RTE Challenge, and BPI[1] test suite.

For three-way task we used the RTE1, RTE2 and RTE3 development sets from Stanford group[2].

Additionally, we generated the following development sets: RTE1+RTE2, RTE2+RTE3, RTE1+RTE3, and RTE1+RTE2+RTE3 in order to train with different corpus and different sizes. In all the cases, RTE4 TAC 2008 gold standard data set was used as test-set.

The remainder of the paper is organized as follows: Section 2 describes the architecture of our system, whereas Section 3 shows the results of experimental evaluation and discussion of them. Finally, Section 4 summarizes the conclusions and lines for future work.

2. System description

This section provides an overview of our system that was evaluated in Fourth Pascal RTE Challenge. The system is based on a machine learning approach for recognizing textual entailment.

In Figure 1 we present a brief overview of the system.

Using a machine learning approach we tested with different classifiers in order to classify RTE-4 test pairs in three classes: entailment, contradiction or unknown.

To deal with RTE4 in a two-way task, we needed to convert this corpus only into two classes: yes and no. For this purpose both contradiction and unknown were taken as class *no*.

There are two variants to deal with every particular text-hypothesis pair or instance. The first way is directly using four features: (1) the Levenshtein distance between each pair, (2) lexical distance based on Levenshtein, (3) a semantic distance based on WordNet and (4) their Longest Common Substring. The second way, is using the “NER-preprocessing module” to determinate whether *non-entailment* is found between text-hypothesis, therefore differing only on the treatment of Named Entities.

The Levenshtein distance [5] is computed between the characters in the stemmed Text and Hypothesis strings. The others three features are detailed below.

Text-hypothesis pairs are stemmed with Porter’s stemmer [3] and PoS tagged with the tagger in the OpenNLP[3] framework.

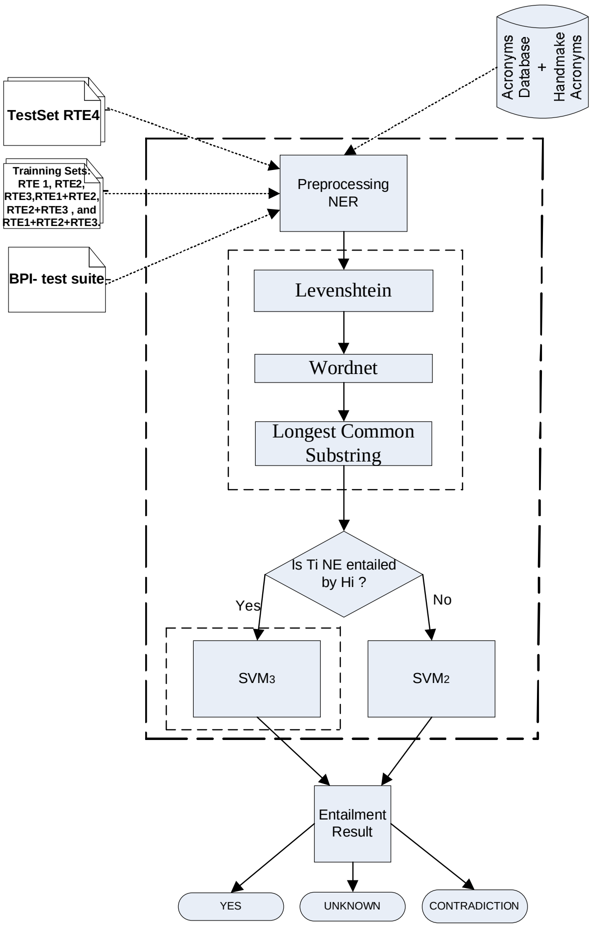


Figure 1.General architecture of our system.

2.1 NER filter

The system applies a filter based on Named Entities. The purpose of the filter is to identify those pairs where the

[1] http://www.cs.utexas.edu/users/pclark/bpi-test-suite/

[2] http://www-nlp.stanford.edu/projects/contradiction/

[3] http://opennlp.sourceforge.net/

system is sure that *no entailment* relation occurs, performing a two steps procedure.

Thus, in the first step the NER-preprocessing module performs NER in text-hypothesis pairs applying several heuristics rules to discard when an entailment relation is not found in the pair. After that, a specialized classifier SVM_2 was trained only with *contradiction* and *unknown* cases of RTE3 corpus and used to classify the pairs between these two classes.

We employed the following the heuristic rules: for each type of Name Entity (person, organization, location, etc.), if there is a NE of this type occurring in H that does not occur in T, then the pair does not convey an entailment and therefore should be classified as either *contradiction* or *unknown*.

The text-hypothesis pairs are tokenized with the tokenizer of OpenNLP framework and stemmed with Porter's stemmer[4] [3]. We also enhanced this NER-preprocess module by using an acronym database [8].

The output module was applied to approximately 10 percent of the text-hypothesis pairs of RTE4. The accuracy of the filter evaluated in TAC'08 was 0.71, with 66 cases correctly classified out of 92 where rules applied.

An error analysis revealed that misclassified cases were indeed difficult cases, as in the following example (pair 807, RTE4):

Text: `Larges scores of Disney fans had hoped Roy would read the Disneyland Dedication Speech on the theme park's fiftieth birthday next week, which was originally read by Walt on the park's opening day, but Roy had already entered an annual sailing race from Los Angeles to Honolulu.`

Hypothesis: `Disneyland theme park was built fifty years ago.`

It was misclassified because of the entity date "fifty years ago" is present in H but not in T. The module unknowns that "fifty years ago" refers to the same date event as "fiftieth birthday".

We plan to extend this module so it can also be used to filter cases where an entailment between text and hypothesis can be reliably identified via heuristic rules.

2.2 Lexical Distance

We use the standard Levenshtein distance as a simple measure of how different two text strings are. This distance quantifies the number of changes (character based) to generate one text string from the other. For example, how many changes are necessary in the hypothesis H to obtain the text T. For identical strings, the distance is 0 (zero).

Additionally, using Levenshtein distance we defined a lexical distance and the procedure is the following:

- Each string T and H are divided in a list of tokens.
- The similarity between each pair of tokens in T and H is performed using the Levenshtein distance.
- The string similarity between two lists of tokens is reduced to the problem of "bipartite graph matching", performed using the Hungarian algorithm over this bipartite graph. Then, we found the assignment that maximizes the sum of ratings of each token. Note that each graph node is a token of the list.

The final score is calculated by:

$$finalscore = \frac{TotalSim}{Max(Lenght(T), Lenght(H))}$$

Where:

TotalSim is the sum of the similarities with the optimal assignment in the graph.

Length (T) is the number of tokens in T.

Length (H) is the number of tokens in H.

2.3 WordNet Distance

WordNet is used to calculate the semantic similarity between a T and an H. The following procedure is applied:

1. Word sense disambiguation using the Lesk algorithm [4], based on Wordnet definitions.

2. A semantic similarity matrix between words in T and H is defined. Words are used only in synonym and hyperonym relationship. The Breadth First Search algorithm is used over these tokens; similarity is calculated by using two factors: length of the path and orientation of the path.

3. To obtain the final score, we use matching average. A bipartite graph is built and computed using Hungarian algorithm.

The semantic similarity between two words (step 2) is computed as:

$$Sim(s,t) = 2 \times \frac{Depth(LCS(s,t))}{Depth(s) + Depth(t)}$$

[4] http://tartarus.org/~martin/PorterStemmer/

Where:

s,t are source and target words that we are comparing (s is in H and t is in T).

Depth(s) is the shortest distance from the root node to the current node.

LCS(s,t):is the least common subsume of s and t.

Finally, the matching average (step 3) between two sentences X and Y is calculated as follows:

$$MatchingAverage = 2 \times \frac{Match\ (X,Y)}{Length\ (X) + Length\ (Y)}$$

2.4 Longest Common Substring

Given two strings, T of length n and H of length m, the Longest Common Sub-string (LCS) method [5] will find the longest strings which are substrings of both T and H. It is founded by dynamic programming.

$$lcs(T,H) = \frac{Length(MaxComSub(T,H))}{\min(Length(T), Length(H))}$$

In all practical cases, min(Length(T), Length(H)) would be equal to Length(H) . Therefore, all values will be numerical in the [0,1] interval.

3. Experimental Evaluation and Discussion of Results

With the aim of exploring the differences between the training sets and machine learning algorithms, we did many experiments looking for the best result to our system.

Thus, we used the following combination of datasets: RTE1, RTE2, RTE3, BPI[5], RTE1+RTE2, RTE1+RTE3, RTE2+RTE3 and RTE1+RTE2+RTE3 to deal with two-way classification task.

In a similar way, we used the following combination of datasets: RTE1, RTE2, RTE3, RTE1+RTE2, RTE2+RTE3, RTE1+RTE3 and RTE1+RTE2+RTE3 of Stanford Group to deal with three-way classification task.

We used five classifiers to learn every development set: (1) Support Vector Machine, (2) Ada Boost, (3) Bayes Net, (4) Multilayer Perceptron (MLP) and (5) Decision Tree using the open source WEKA Data Mining Software [7]. In all the tables results we show only the accuracy of the best classifier.

The RTE4 data set is three-way. Nevertheless, this corpus was converted into “RTE4 2-way” taking *contradiction* and *unknown* pairs as no- entailment in order to test the system in the two-way task.

Our results for RTE two-way classification task are summarized in Table 1 below. In addition, table 2 shows the results obtained in RTE three-way classification task.

Dataset	Classifier	Acc %
RTE3	**MLP**	**58.4%**
RTE3 With NER Module	SVM	57.6%
RTE2 + RTE3	MLP	57.5%
RTE1 + RTE2 + RTE3	MLP	57.4%
RTE1+ RTE3	Decision Tree	57.1%
RTE1 + RTE2	Decision tree	56.2%
RTE2	ADA Boost	55.6%
	Decision tree	55.6%
	Bayes Net	55.6%
RTE1	ADA Boost	54.6%
	Bayes Net	54.6%
Baselines	-	50%
BPI	BayesNet	49.8%

Table 1.Results obtained in two-way classification task.

Dataset	Classifier	Acc %
RTE3	**MLP**	**55.4%**
RTE1 + RTE3	MLP	55.1%
RTE1 + RTE2 + RTE3	MLP	54.8%
RTE1 + RTE2	SVM	54.7%
RTE2	SVM	54.6%
RTE2+RTE3	MLP	54.6%
RTE1	SVM	54%
RTE3-With NER Module	SVM	53.8%
Baseline	-	50%

Table 2.Results obtained in three-way classification task using Stanford datasets.

Here we noted that using RTE3 instead of RTE2 or RTE1 in both classification tasks (two and three way) always achieves better results. Interestingly, the RTE3 training set alone outperforms the results obtained with any other combination of RTE-s datasets, even despite the size of increased corpus. Thus, for training purpose, it seems that any additional datasets to RTE-3 introduces "noise" in the classification task.

[5] http://www.cs.utexas.edu/users/pclark/bpi-test-suite/

(Zanzotto et al) shown that RTE3 alone could produce higher results that training on RTE3 merged with RTE2 for the two-way task. Consequently, it seems that *it is not always true that more learning examples increase the accuracy* of RTE systems. These experiments provide additional evidence for both classification tasks. However, this claim is still under investigation.

Always the RTE1 dataset yields the worse results, maybe because this dataset has been collected with different text processing applications (QA, IE, IR, SUM, PP and MT), and our system do not have into account it.

In addition, a significant difference in performance of 3.8% and 8.6% was obtained using different corpus, in two-way classification task (with and without the BPI development set, respectively).

The best performance of our system was achieved with Multilayer Perceptron classifier with RTE-3 dataset; it was 58.4% and 55.4% of accuracy, for two and three way, respectively.

The average difference between the best and the worst classifier of all datasets in two way task was 1.6%, and 2.4% in three-way task.

On the other hand, even if the SVM classifier does not appear as 'favorite' in neither classification task, in average SVM is one of the best classifiers.

We have to remark that in two-way task we obtained a difference of 3.8% between the best and worst combination of datasets and classifiers; meanwhile, in three-way task a slight and not statistical significant difference of 1.4% between the best and worst combination of datasets and classifiers is found. So, it suggests that the combination of data set and classifier has more impact over 2-way task than over 3-way task.

The performance in all the cases was clearly above those baselines. Only using BPI in two-way classification we obtained a worse result than baseline, and it is because BPI is syntactically simpler than PASCAL RTE; therefore, it seems that is not good enough training set for machine learning algorithm.

Although the best results were obtained without using the Name Entity Preprocessing module, we believe these results could be enhanced. The accuracy of this module was 71%, but the misclassified instances provide evidence that could be improved almost up to 80% (e.g: improving the acronym database), and having into account the coverage of corpus that was 10%, it could impact positively on the overall performance of the system. While this Name Entity Preprocessing module approach performed reasonably well in these evaluations, we feel that even better results could be obtained by adding heuristic rules and knowledge base information.

With the aim of analyzing the feature-dependency, we calculated the correlation of them. The correlation and causation are connected, because correlation is needed for causation to be proved.

The correlation matrix of features is shown below:

Features	**1**	**2**	**3**	**4**
1	-	0,8611	0,6490	0,2057
2	0,8611	-	0,6951	0,0358
3	0,6490	0,6951	-	0,1707
4	0,2057	0,0358	0,1707	-

Table 3.Correlation matrix of features.

The table shows that features (1) and (2) are strongly correlated, so we experimented eliminating feature (1) to assess the effect on the overall performance over cross validation, and we obtained that accuracy slight decreases in 1%. Similar results are obtained by eliminating feature (2).

Additionally, we calculated the Kappa statistics over all development set using WEKA (Witten and Frank, 2000) for both 2-way and 3-way task classification. The average for Kappa measure was 0.138 for two-way task and 0.168 for three-way task.

In general, because of the corpus was incremented we obtained better values for Kappa. Nevertheless, the best value was obtained with RTE-3 two ways using MLP. In this case, the Kappa measure was 0.35 for cross validation experiment (See Tables 4 and 5).

There are two main reasons because of the values were slight: the size of the corpus and the mistakes made in the class contradiction, which was the most difficult class to predict in the 3-way classification.

Finally, we assessed our system using cross validation technique with ten folds to every corpus, testing over our five classifiers for both classification tasks.

The results are shown in the tables 4 and 5 below.

Dataset	**Classifier**	**Accuracy%**
RTE3	MLP	**65.5%**
RTE2 + RTE3	MLP	60.68%
RTE1+RTE2+RTE3	MLP	59.35%
RTE2	SVM	56.62%
RTE1+RTE2	SVM	55.84%
RTE1	Decision tree	54.70%

Table 3.Results obtained with Cross Validation in three-way task.

Dataset	Classifier	Accuracy%
RTE3	BayesNet	**67.85%**
BPI	BayesNet	64%
RTE1 + RTE2 + RTE3	MLP	63.16%
RTE2	SVM	60.12%
RTE1+RTE2	MLP	59.79%
RTE1	SVM	57.83%

Table 4.Results obtained with Cross Validation in two-way task.

The results on test set are worse than those obtained on training set, which is most probably due to the overfitting of classifiers and because of the possible difference between these datasets.

4. Conclusion and Future Work

We presented our RTE system that is based on a wide range of machine learning classifiers. It was a workbench that gave us a vision and knowledge about the structure of the data set and the abilities of different classifiers to learn them.

As a conclusion about development sets, we mention that the results performed using RTE3 were very similar to those obtained by the union of the RTE1 + RTE2+RTE3 for both 2-way and 3-way tasks. Thus, the claim that *using more training material helps* seems not to be supported by these experiments.

Additionally, we concluded that the relatively similar performances of RTE3 and RTE3 with NER preprocessing module suggests that further refinements over heuristic rules can achieve better results.

Despite not presenting an exhaustive comparison among all available datasets and classifiers, we can conclude that the best combination of RTE-s datasets and classifiers chosen for two way task produce more impact that the same combination for three way task, almost for all experiments that we did. In fact, the use of RTE3 alone improved the performance of our system.

Finally, we conclude that RTE3 corpus for both two and three way outperforms any other combination of RTE-s corpus using Multilayer Perceptron classifier.

Future work is oriented to experiment with additional lexical and semantic similarities features and test the improvements they may yield. Additional work will focused on improving the performance of our NE preprocessing module.

5. References

[1] Prodromos Malakasiotis and Ion Androutsopoulos. *Learning Textual Entailment using SVMs and String Similarity Measures.* ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, (ACL 2007), Prague, Czech Republic, 2007.

[2] Julio Javier Castillo, and Laura Alonso i Alemany. *An approach using Named Entities for Recognizing Textual Entailment.* TAC 2008, Gaithersburg, Maryland, USA, November 2008.

[3] M. Lesk. *Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone.* In SIGDOC ’86, 1986.

[4] Gusfield, Dan. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* CUP, 1999.

[5] V. Levenshtein. *Binary Codes Capable of Correcting Deletions, Insertions and Reversals.* Soviet Physics Doklady, 10:707, 1966.

[6] Ian H. Witten and Eibe Frank (2005). *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[7] Alvaro Rodrigo, Anselmo Peñas, Jesus Herrera, Felisa Verdejo. *Experiments of UNED at the Third RTE* Challenge. Proceedings of the ACL-PASCAL 2007.

[8] British Atmospheric Data Centre (BADC) acronym database: *http://badc.nerc.ac.uk/help/abbrevs.html*

[9] D. Inkpen, D. Kipp and V. Nastase. *Machine Learning Experiments for Textual Entailment.* Proceedings of the second RTE Challenge, Venice-Italy, 2006.

[10] Bill Dolan, Chris Quirk, and Chris Brockett. 2004. *Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources.* In COLING ’04: Proceedings of the 20th international conference on Computational Linguistics, page 350, Morristown, NJ, USA. Association for Computational Linguistics.

[11] F. Zanzotto, Marco Pennacchiotti and Alessandro Moschitti. *Shallow Semantics in Fast Textual Entailment Rule Learners*. In Proceedings of the Third Recognizing Textual Entailment Challenge, Prague, 2007.

[12] Marie-Catherine de Marneffe, et al. Manning.*Learning to distinguish valid textual entailments.* In Proceedings of the Third Recognizing Textual Entailment Challenge, Italy, 2006.

LOGICON: A System for Extracting Semantic Structure using Partial Parsing

Kais Dukes
School of Computing, University of Leeds
LS2 9JT, United Kingdom
sckd@leeds.ac.uk

Abstract

Partial parsing is an established NLP technique used to perform syntactic analysis without generating a full constituent parse tree. This paper presents LOGICON, an end-to-end system using partial parsing, which assigns novel semantic structures to natural language text. Evaluating against a test set of 500 previously unseen sentences, the system has an accuracy of 62.4% as measured by exact matching against the expected semantic output. Since partial parsing is used, the system is robust and will assign partial semantic structure to sentences it may not fully understand. As stochastic methods are not used, the system is deterministic and fast. A syntactic tagging scheme is proposed which is closely aligned to the corresponding semantics. The system was developed as part of a PhD research project, and was written to evaluate partial parsing as the first step to creating a full natural language question-answering system.

Keywords

natural language processing, partial parsing, annotated corpora, constituent structure, thematic relations, semantic role labeling, syntax parse trees, part-of-speech tagging

1. Introduction

The LOGICON system was developed as part of an ongoing PhD research project, with the aim of using partial parsing to extract semantic structures from natural language. LOGICON contrasts with PARASITE, a system which produces formal semantics for unrestricted text [9], since partial parsing [1][2] is used instead of deep parsing, and semantic roles are used for representing meaning as opposed to logic statements with variables and quantifiers.

For example, given a simple sentence focusing around an event, LOGICON attempts to identify roles for the *actor* (who did the event), the *action* (what the event was) and the *target* (what entity the actor performed the action on). The system employs simple partial parsing techniques as described by Abney [1], [2]. A syntax-driven approach is then used to derive semantic roles through recursion.

2. Semantic Structures

Thematic relations are an intuitive approach to assigning meaning to the constituents of a sentence. A typical problem is what role to assign to a noun phrase. Traditionally thematic relations include Agent, Patient, Theme, Location, etc. However, there is no definitive list of roles, and in some cases which role to use is not immediately clear: in "the key opened the door" is the key the agent or the instrument? In order to produce a practical system, the research focused on working with a small set of well-defined roles:

Table 1. Semantic relations describing an event

Relation	Description
ACTION	the action, or main verb
ACTOR	the doer of the action
TARGET	what the action was performed on
LOCATION	where the action was performed
TIME	when the action was performed

The ACTOR role is typically assigned to the syntactic subject of the sentence, and the TARGET is typically assigned to the object. Together, the roles are grouped into a semantic structure called an EVENT. For the simple sentence "Jack helped John", the corresponding semantic structure produced by LOGICON would be:

```
EVENT:
    ACTOR: Jack
    ACTION: help
    TIME: PAST
    TARGET: John
```

A semantic structure called a LINK is used to represent sentences that use a copula to link a subject (the SOURCE) to its predicate (the TARGET). For example, "The apple is red" would have the following semantic structure:

```
LINK:
    SOURCE: apple
    TARGET: red
```

Table 2 below gives a brief summary of the important keywords used in the semantic structures found in the annotated corpus:

Table 2. Keywords in LOGICON semantic structures

Keyword	Description
SPEAKER	represents the first person
LISTENER	represents the second person
OTHER	represents the third person
OF	used in a possessive construction
CONFIRM	"*Is* the sky blue?"
EXPLAIN	"*Why* is the sky blue?"
QUANTIFY	"How much" / "How many"
PARAMETER	second object (ditransitive verbs)
LOCATION	"Jack put the book *on the table*"
SPECIFIC	represents the definite article
GENERAL	represents an indefinite article
CONCEPT	an isolated noun phrase
POSSIBLE	"Jack *might* eat"
EXPECTED	"Jack *will* eat"
RECOMMENDED	"Jack *should* eat"
MODIFIER	"Jack ate *quickly*" (adverb)
NOT	used to negate a structure
AND	"Jack *and* Mary are clever"
OR	"Eat the apple *or* the orange"

Special handling is given to pronouns and to possessive constructions, using the first 4 keywords listed in table 2. As an example, LOGICON translates the sentence "You broke my car" into the corresponding semantic structure:

```
EVENT:
    ACTOR: LISTENER
    ACTION: break
    TIME: PAST
    TARGET: car OF SPEAKER
```

The aim of the system is to produce enough semantic detail to enable effective question-answering. Since partial parsing, and not deep parsing is used, some structures are not dissected. For example the internals of noun phrases are not handled directly.

In this respect, the semantic structures can be considered a form of semantic role labeling. The structures are general-purpose so that it would be more accurate to say that they are an intermediate form between the frame semantics found in FrameNet [3] and the verb-argument annotations found in the PropBank corpus [8].

2.1 Semantic recursion

Since natural language is inherently recursive, it is not unreasonable to expect that any corresponding semantic structures should show similar recursion. The semantic structures generated by LOGICON are syntactically-driven, that is to say they are derived directly from parse trees constructed via partial parsing. Since these parse trees are recursive structures, so are the corresponding semantics.

A simple example would be the sentence "Who said time is money?". In the corresponding semantic structure, this is analyzed as a LINK (a subject/predicate construction) embedded within an EVENT (an action in time or space):

```
EVENT:
    ACTOR: UNKNOWN
    ACTION: say
    TIME: PAST
    TARGET: LINK:
                SOURCE: time
                TARGET: money
```

The actor (the doer of the action) is UNKNOWN ("who?"). The UNKNOWN keyword is used as a placeholder for thematic roles in sentences which use interrogative pronouns. In the event structure above, the target of the action (what was said) is itself another semantic structure, a link between a subject and its predicate: "time *is* money".

3. Partial Parsing

3.1 Abney’s partial parsing scheme

The partial parsing scheme introduced by Abney [1] and implemented in the Cass partial parser [2], successively builds a parse tree bottom-up by using a cascade of finite state transducers. Customizable patterns are used to define the regular expressions used to parse at each level. These patterns are specified in a human readable format (similar to Backus-Naur form) and are then complied into a unified finite state transducer automatically. A distinguishing feature of the original scheme is that there is no definite top-level node representing the entire sentence. The system is more like a chunking analyzer as opposed to a full syntactic parser.

The main advantages of the Cass partial parsing scheme is that it is robust (it will not fail to produce a partial analysis given input it may not fully understand), it is fast (orders of magnitude faster than stochastic parsers) and relatively easy to implement.

3.2 The annotated corpus

An annotated corpus was constructed at the start of the project. By adopting a corpus-driven methodology, the effectiveness of potential parser rules was decided by available corpus evidence. The annotations were produced as follows:

1. A set of 2000 sentences was collected.
2. For each sentence, the semantic structure expected to be produced by the system was manually annotated.
3. From the expected semantic structure, a syntactic parse tree was also annotated that would provide the suitable semantic skeleton from which to derive the semantics.

After annotation the corpus was divided into two sets: a training set of 1500 sentences, and an evaluation set of 500 sentences. The training set would be used as a reference when building the system, in order to test the effectiveness of the parser during its construction, and to try out various partial parsing rules. An example annotated sentence is shown below:

"Who wrote ‘The Moon is a Harsh Mistress?’"

```
(EV
    (C Who) (V wrote)
    (LN
        (C (Q The) (C Moon))
        (AUX is)
        (C (Q a) (C Harsh Mistress))))
```

```
EVENT:
    ACTOR: UNKNOWN
    ACTION: write
    TIME: PAST
    TARGET: LINK:
                SOURCE: SPECIFIC Moon
                TARGET: GENERAL Harsh Mistress
```

The following resources were used to construct corpus:

1. Example sentences from the Link Grammar Parser [10].
2. Sentences based on patterns in A.L.I.C.E. [11].
3. Questions from the TREC-10 QA track [6].
4. Sentences from novels in Project Gutenberg.
5. News headlines from news.google.com.

Different sources were used so that a wide-coverage parser could be constructed. Focusing on a particular genre – such as newspaper text – might have resulted in a more limited parser. A large proportion of the data was derived from the question templates found in the A.L.I.C.E. chat program, which is relevant to question-answering because this data was formed after studying the most frequent inputs given to a popular chat system.

3.3 Partial parsing in LOGICON

Three possible parsing schemes were considered at the outset of the project: using a stochastic parser such as Bikel’s parser [4], using a dependency parser such the Link Grammar Parser [10], or using a partial parser. An existing stochastic parser was not suitable for use in LOGICON, because either these are pre-trained on a different tagset or need to be trained using a large corpus. It was felt that converting the output from the Link Grammar Parser would be too time consuming, so it was decided to construct a partial parser which matched the syntax in the annotated corpus. A Brill tagger using transformation-based machine learning was first applied to the training set [5].

With an effective part-of-speech tagger in place, Abney’s original partial parsing scheme was then adapted. Initially, this yielded encouraging results. Out of the 1500 sentences in the training corpus, 7 simple rules resulted in partial syntax trees which had an accuracy of 90.84% as measured by the number of nodes parsed and connected to the correct constituent nodes. A total of 35 rules were finally used.

3.4 The partial parsing algorithm

The partial parsing algorithm used by the LOGICON system is described as follows[1]. The parser constructs a parse tree bottom-up. At each stage of its operation, there is a set of top-level nodes, which are grouped together to form new top-level nodes at the next iteration. The result of the algorithm is a partial parse tree, defined as a set of one or more final top-level nodes, each of which is a complete parse tree:

1. Construct a node for each word, using part-of-speech tags from the tagger.
2. For each parser rule R, apply R to the top-level nodes.
3. If at least one rule did apply, repeat step 2 until no rules apply, and no new top-level nodes can be created.

Figure 1. Rule in partial parser specifying a new node

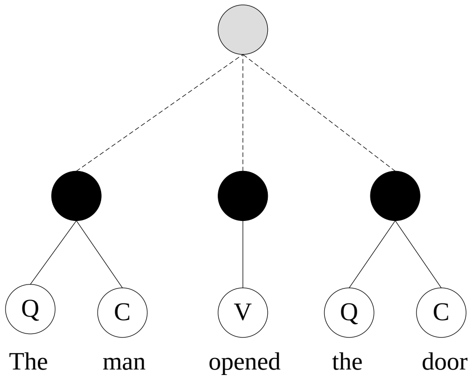


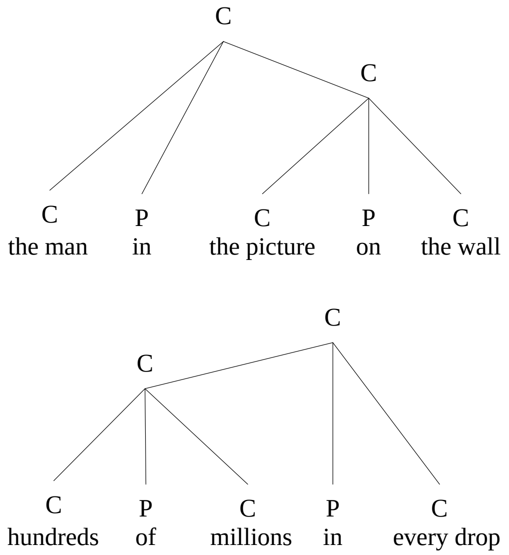
Figure 1 above shows a syntactic structure constructed by the parser during its analysis. The current top-level nodes are shown in black. At this stage of operation, the three top-level nodes represent a noun-phrase, followed by a verb, followed by a second noun phrase. These would be recognized by a parser rule as a subject-verb-object construction, and these 3 nodes would be collected into a new top-level node shown in gray.

[1] See the appendix for a description of the partial parsing algorithm in pseudo-code

The ordering of rules in the parser is crucial, as this represents rule precedence. The rules which operate on lower parts of the parse tree are listed first. Rules which operate on similar syntactic structures are listed together. However they are ordered so that the rule which is more applicable in general will operate first, to create new top-level nodes with the correct precedence.

Figure 2 below shows two sentences with the same part-of-speech tags, but with different constituent structure. In the annotated corpus, two noun phrases separated by a preposition (e.g. 'the picture *on* the wall') are grouped together into a symmetric compound noun phrase. This analysis is used to support the expected semantic structure. Ambiguity arises when the parser is faced with the sequence C + P + C + P + C. As figure 2 indicates, this can be analyzed in one of two ways.

Figure 2. Ambiguity in preposition and noun phrase structure



The first right-to-left grouping was found to be more common in the corpus. The rule which deals with prepositions builds this structure by default. The second grouping also occurs, and in this case the behavior is overridden by adding lexical information to the parser rule.

3.5 The tagging scheme

It was decided to keep the tagging scheme used by LOGICON to be as close as possible to the final semantic structure. A sample sentence found in the training corpus is: "Who was the lead singer for the Commodores?". This is analyzed as a copula link, with the subject 'who'. The predicate is a compound phrase with a preposition linking two noun phrases ('the lead singer' and 'the Commodores'). The syntactic structure constructed via partial parsing is:

```
(LN
    (C Who) (AUX was)
    (C
        (C (Q the) (C lead singer))
        (P for)
        (C (Q the) (C Commodores))))
```

A reduced tagset of 18 tags is used by the partial parser:

Table 3. Semantically-aligned tagset used for syntax tree nodes

Tag	Description or example
N	noun
V	verb
AUX	auxiliary verb
P	preposition
Q	quantifier / determiner
SYM	symbol / punctuation
NEG	negation ("no, not")
POS	possessive ("Jack’s house")
CONJ	conjunction ("and, or")
T	time phrase ("2pm")
LOC	location phrase ("in London")
COMP	complementizer / relative pronoun
C	concept / noun phrase
EV	subject-verb-object event
LN	subject-copula-predicate link
XL	explanation question ("why")
MOD	modifier (adverb)
OP	mathematical operation ("2 + 2")

3.6 Mapping partial parsing to semantics

The translation algorithm is a recursive map. The translator accepts as input the results of the partial parser and performs a recursive algorithm which visits each node in turn, bottom-up. A sequence of semantic translation rules are applied, with each rule operating on a pre-defined syntactic tag. Thematic relations are deduced directly from the syntax, designed with semantic analysis in mind. For example, rules will map a verb tag to an ACTION role, and a time tag to a TIME role. Since a parent node’s children will already have semantics attached, these structures can be used to construct the next-level of analysis, and so on, until the entire partial tree has semantics attached to each node. The final output of the LOGICON system is the resulting semantic structure attached to the top-level nodes.

4. Evaluation

4.1 Evaluation against the annotated corpus

The annotated corpus was divided into a training set of 1500 sentences and an evaluation set of 500 sentences. When applied to the evaluation set, the system produced the exact expected semantic output for 312 of the 500 sentences, giving an accuracy score of 62.4%.

Table 4. Evaluation matching against exact expected semantics

Matched	Not matched	Total	%
312	188	500	62.4

A qualitative analysis of the errors indicated that most of the inaccuracies were due to the part-of-speech tagger. The evaluation set contained words not previously seen by the Brill tagger which resulted in incorrect parts-of-speech.

4.2 Parsing speed

Despite the lower than expected accuracy, the system did demonstrate a good trade-off between speed versus depth of analysis. The algorithm presented is wholly deterministic, making no use of stochastic techniques or backtracking. The entire 2000 sentences took 0.75 seconds to process. This measurement was the average of several runs on an Lenovo T61 Laptop, running two Intel Core Duo processors, at 2.4 GHz.

5. Conclusions and Future Work

In this paper the LOGICON system was presented and a novel set of semantic structures were described, driven by syntax. The system employs partial parsing techniques using a tagging scheme in which syntax and semantics are closely aligned. With a few partial parsing rules, reasonable accuracy was obtained. Future work will involve refining the parser using a more accurate part-of-speech tagger such as SVMTool [7] and then applying the system to the question-answering domain.

6. Appendix: Parsing Algorithm

The algorithm used by the LOGICON partial parser is shown in pseudo-code below:

```
- create initial nodes from part-of-speech tags
- repeat until no rule applies:
    - for each parser rule R:
        - repeat while R applies to existing top-level nodes:
            - use R to create a new top-level node
```

In the training corpus of 1500 sentences, there were a total of 6722 constituent nodes. The second column in table 5 shows the number of nodes generated by each parser rule. The fourth column shows the cumulative percentage. With a few general rules reasonable accuracy can be achieved, but producing increased accuracy from the parser requires writing a larger number of specialized rules.

Table 5. Top 10 most common partial parsing rules

Parser rule	Nodes	%	Cum. %
C → N+	2104	31.30	31.30
C → C + Q	1104	16.42	47.72
C → named entity	1049	15.61	63.33
EV → contains V	761	11.32	74.65
LN → contains AUX + C	550	8.18	82.83
C → C + P + C	362	5.39	88.22
V → P + V	176	2.62	90.84
LN → contains AUX	138	2.05	92.89
C → C + POS + C	96	1.43	94.32
AUX → AUX + NEG	81	1.20	95.52

7. References

[1] S. Abney. Partial Parsing via Finite-State Cascades, in Workshop on Robust Parsing (ESSLLI), 1996.

[2] S. Abney. The SCOL Manual, http://www.vinartus.net/spa, 1997.

[3] C. Baker, C. Fillmore, J. Lowe. The Berkeley FrameNet project, in Proceedings of COLING/ACL, 1998.

[4] D. Bikel. Intricacies of Collins' Parsing Model, Computational Linguistics, 30(4), 2004.

[5] E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging, Computational linguistics, 1995.

[6] J. Chen, A. Diekema, et al. Question answering: CNLP at the TREC-10 question answering track. In Proceedings of the Tenth Text REtrieval Conference, 2001.

[7] J. Gimenez, L. Marquez. SVMTool: A general POS tagger generator based on Support Vector Machines. In Proceedings of the 4th International Conference on Language Resources and Evaluation, 2004.

[8] M. Palmer, G. Daniel, K, Paul. The proposition bank: An annotated corpus of semantic roles, Computational Linguistics, 31(1), 2005.

[9] A. Ramsay, H. Seville. Models and Discourse Models, Journal of Language and Computation, 1(2), 2000.

[10] D. Sleator, D. Temperley. Parsing English with a Link Grammar, Third International Workshop on Parsing Technologies, 1993.

[11] R. Wallace. The Annotated A.L.I.C.E. AIML, http://www.alicebot.org/aiml/aaa, 2007

Framework for using a Natural Language Approach to Object Identification

Mosa Emhamed Elbendak
Northumbria University.
Camden Str.
Newcastle upon Tyne, NE2 1XE
mosa.elbendak@unn.ac.uk

Abstract

Object-oriented analysis and design has now become a major approach in the design of software system. This paper presents a method to automate natural language requirements analysis for object identification and generation based on the Parsed Use Case Descriptions (PUCDs) for capturing the output of the parsing stage. We employ Use-Case Descriptions (UCDs) as input into the whole framework of identification of classes and relationship. PUCD is used to extract nouns, verbs, adjectives and adverbs from use case descriptions as part of an identification process to identify objects/classes and relationships. We refine classes by human expert to produce a class model as output.

Keywords

Requirements specifications, object-oriented, parsing, analysis, Natural Language Processing.

1 Introduction

The process of requirements identification is considered one of the most critical and difficult tasks in database design because most of the input to this process is in natural languages, such as English, which are inherently ambiguous. Developers need to interact with users in their language. Also they need to review and analyse documents written in natural language. This paper presents the work we have achieved so far to find a solution to the problem of automatic identification of objects/classes and relationships from a Requirements Specifications (RS) written in a natural language. Firstly, we review existing literature on the subjects of requirements specifications, object identification, and conceptual database design. The different techniques adopted in the natural language processing systems that attempt to transform natural language to conceptual models have been reviewed. Moreover, we have reviewed the rules to convert English sentences into ER and EER diagrams to determine entity types, attribute types and relationship types.

We intend to employ use-case descriptions as input into the whole framework of identification of classes and relationships, using an existing parsing tool to identify noun phrases, verb phrases, adjectives and adverbs. We propose an intermediate representation called Parsed Use Case Descriptions (PUCD) for capturing the output of the parsing stage, which is then used in subsequent steps. A PUCD is a set of original sentences, parsed sentences, nouns, verbs, adjectives and adverbs, which we use to extract nouns, verbs, adjectives and adverbs from use case descriptions. The next step is the identification process to identify objects/classes, attributes, operations, associations, aggregations and inheritance so as to produce a class model. We refine classes by human expert. In addition to the literature review, the work achieved to date includes an outline of the proposed method for object identification, which is based on existing work on how to map English sentences into conceptual models. The next step identify objects/classes, attributes, operations, and the association, aggregation and inheritance abstractions to produce a class model by applying a set of rules.

2 Motivation

Our motivation is therefore to identify automatically objects/classes and relationships from requirements specifications written in a natural language, e.g. English, so as to increase efficiency in the use of scarce resources and to reduce errors in dealing with complex requests. Therefore we investigate how natural language processing tools and techniques can be used to support the Object-Oriented Analysis (OOA) process. We assume that an English description of the software problem to be solved has already been written. This can be an initial description of the problem or a more detailed list of requirements. We employ Use Case Descriptions (UCDs) as input for identifying classes and relationships as these are well-structured texts . Automation of the Systems Development Life Cycle (SDLC) can alleviate the critical problems of ambiguity, inconsistencies and conflicts in functional requirements [11]. Use case descriptions are very effective in analyzing and capturing functional requirements. They play a major role in defining the processes and actors who can be part of the shareholders of the system. They can be extended, automated and implemented to achieve complete, consistent, and conflict free requirements specification.

The contribution of this paper is to develop a PUCD automatically from use case descriptions as input to extract nouns, verbs, adjectives and adverbs.

3 Challenges

Handling of Cycles. Software engineering tools nearly always involve a number of cycles with discussion at the end of each iteration on each solution. Thus we have the well-known first- and second-cut approaches. This style of working facilitates the refinement of the models developed. In the work described here we anticipate that two or three cycles will be needed to optimise convergence on an agreed outcome. A framework has to be constructed for handling the cycles.

Use of Natural Language (NL). There are many difficulties often associated with the use of NL which can be summarised as following:

- The ambiguity and complexity of NL are major problems in requirements specifications, as they may lead to misunderstanding between the different users, which most likely will badly affect customer satisfaction with the implementation produced. Furthermore any errors, mistakes or inconsistencies incurred at this stage can be very costly later especially when a system has already been implemented. It has been reported that the cost difference to correct an error in the early stage compared with leaving it till the end is 1:100. See [18, 2, 13].

- The analysis process is considered to be one of the most critical and difficult tasks because most of the input to this process is in natural language such as English.

- Automatic identification of objects/classes and relationships is potentially faster than manual identifications but may be less accurate.

- There is no standard method for automatically identifying objects and classes from English sentences.

- With NLP it is now possible to distinguish nouns and verbs but it is not so easy to classify the verbs as particular types of relationships such as association, aggregation or inheritance, in the identification process.

- There is little or no adaption of standards like UML for expressing requirements specification (RS) for the purpose of object identification.

4 Background and Related Work

Previous studies provide some rules for mapping natural language elements to object-oriented concepts. However, it appears that the coverage is incomplete. For example Abbott [1] first suggested that nouns indicate classes and objects, while verbs can denote behaviours. Researchers and software designers such as Booch et al., and Liang et al., [3, 12] have come to the conclusion that object identification and the refinement process are an ill-defined task, because of the difficulty of heuristics and the lack of a unified methodology for analysis and design. This is mainly due to the lack of a formalism for object-oriented analysis and design.

Although there are many projects focusing on Computer Aided Software Engineering (CASE) tools for object-oriented analysis and design, there are only a few focusing on the formalisation and implementation of the methodology for the object model creation process. Also they are not developed well for the software design that requires collaborative working among members of a software design project team. Wahono and Far [21, 20] examine the issues associated with the methodology for collaborative object-oriented analysis and design. This system is called OOExpert.

Data Model Generator (DMG) is a rule-based design tool by Tjoa and Berger [19] which maintains rules and heuristics in several knowledge bases and employs a parsing algorithm to access information on a grammar using a lexicon designed to meet the requirements of the tool. During the parsing phase, the sentence is parsed by retrieving necessary information using the rules and heuristics to set up a relationship between linguistic and design knowledge. The DMG has to interact with the user if a word does not exist in the lexicon or the input of the mapping rules is ambiguous. The linguistic structures are then transformed by heuristics into EER concepts. Though there is a conversion from natural language to EER models, the tool has not yet been developed into a practical system.

ER generator by Gomez et al. [8] is another rule-based system that generates E-R models from natural language specifications. The E-R generator consists of two kinds of rules: specific rules linked to semantics of some words in sentences, and generic rules that identify entities and relationships on the basis of the logical form of the sentence and of the entities and relationships under construction. The knowledge representation structures are constructed by a Natural Language Understanding (NLU) system which uses a semantic interpretation approach.

CM-Builder by Harmain and Gaizauskas [9] is a natural language based CASE tool which aims at supporting the analysis stage of software development in an object-oriented framework. The tool documents and produces initial conceptual models represented in the Unified Modelling Language. The system uses discourse interpretation and frequency analysis in linguistic analysis. For example, attachment of postmodifiers such as prepositional phrases and relative clauses is limited. Other shortcomings include the state of the knowledge bases which are static and not easily updateable nor adaptive. Meziane and Vadera [15] and Farid [7] implemented a system for the identification of VDM data types and simple operations from natural language software requirements. The system first generates an Entity-Relationship Model (ERM) from the input text followed by VDM data types from the ERM.

Mich and Garigliano [17] and Mich [16] described an NL-based prototype system, NL-OOPS, that is aimed at the generation of object-oriented analysis models from natural language specifications. This system demonstrated how a large scale NLP system called LOLITA can be used to support the OO analysis stage.

Some researchers, also advocating NL-based systems, have tried to use a controlled subset of a natural language to write software specifications and build tools that can analyse these specifications to produce useful results. Controlled natural languages are developed to limit the vocabulary, syntax and semantics of the input language. Macias and Pulman [14] discuss some possible applications of NLP techniques, using the CORE Language Engine, to support the activity of writing unambiguous formal specifications in English.

The research work described above has provided valuable insights into how NLP can be used to support the analysis and design stages of software development. However, each of these approaches has weaknesses, which means that as yet NL-based CASE tools have not emerged into common use for OO analysis and design. Abbott and Booch's work describes a methodology, but they have not produced a working system which implements their ideas. Meziane produced workable systems but these required an unacceptable level of user interaction such as accepting or rejecting noun phrases to be represented in the final model on a sentence by sentence basis as the requirements document is processed.

Mich and Garigliano's approach, which is closest to our own, is reliant on the coverage of a very large scale knowledge base and the impact of (inevitable) gaps in this knowledge base on the ability of the system to generate usable class models is unclear. It is also worth noting that none of these systems, so far as we are aware, has been evaluated on a set of previously unseen software requirements documents from a range of domains. This ought to become a mandatory methodological component of any research work in this area, as it has in other areas of language processing technology, such as the DARPA-sponsored Message Understanding Conferences (see, e.g. [10]).

5 Research Method

Figure 1 outlines the research activities involved in the research method as follows. The method begins with the Requirements Specification (RS) that describes the structure and behaviour of the system. RS are usually written in Natural Language (NL) e.g., English. NL enables non-technical users to understand the requirements. NL needs to be analyzed, transformed and restructured into a form used as a notation for software requirements specifications. NL includes text from different linguistic levels such as words, sentence and meaning.

The object identification process uses RS to identify fundamental elements (e.g., classes, attributes, operations and relationships) of a conceptual design. The process to generate a conceptual model uses a diagrammatic notation (e.g., UML class diagram or ERM). Figure 1 appears to be recursive but in practice three cycles are likely to be sufficient. The purpose of the first cycle is to check if there is any error in the requirements specifications; the purpose of the second cycle is to correct the errors by refinement to the requirements specification; the purpose of the third cycle completes the whole process by a verification and validation process which checks whether the class model conforms to the original RS. This model therefore includes object identification, conceptual design generation, refinement and verification and validation.

The work described here will make a greater use of automation than earlier approaches, by considering a generated structured output PUCD automatically. The automation is assisted by the sentence-based nature of the parser and the use of a complex set of rules to assist with object identification.

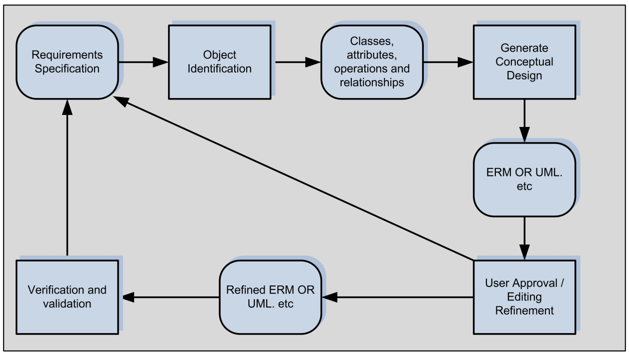

Requirements Specification
Object Identification
Classes, attributes, operations and relationships
Generate Conceptual Design
ERM OR UML. etc
User Approval / Editing Refinement
Refined ERM OR UML. etc
Verification and validation

Fig. 1: *Outline of research method*

6 Overview of the Proposed Method

Figure 2 gives an overview of the identification of classes and relationship. Use Case Descriptions (UCDs) represent input to the process as a whole. As explained in more detail later, the parsing process as a whole involves as preliminaries a tokenizer, sentence splitter, part-of-speech tagger and chunking, followed by the parser itself. The PUCD generated is a set of original sentences, parsed sentences, nouns, verbs, adjectives and adverbs. A preliminary class model is generated from the PUCD, which is then refined by a human expert. The steps to design the class diagram from NL are listed below:

Step 1: Parse the use-case description(s) using Memory-Based Shallow Parser (MBSP) to generate noun phrases and verb phrases.

Step 2: Generate PUCD from the output of step 1.

Step 3: Identify classes, and association, aggregation and inheritance abstractions from PUCD objects/classes, using the rules to produce a class model.

Step 4: Refine the output of step 3 using a human expert.

6.1 Use Case Descriptions (UCDs)

UCDs written in a natural language are usually employed for specifying of functional requirements. The format of a use case is not standardized. UCD takes a function requirements text file containing the software requirements. We impose no limitations on the form of the requirements document provided that it is written in English. It can be in the structure of a general problem statement describing the software problem or a list of more detailed functional requirements.

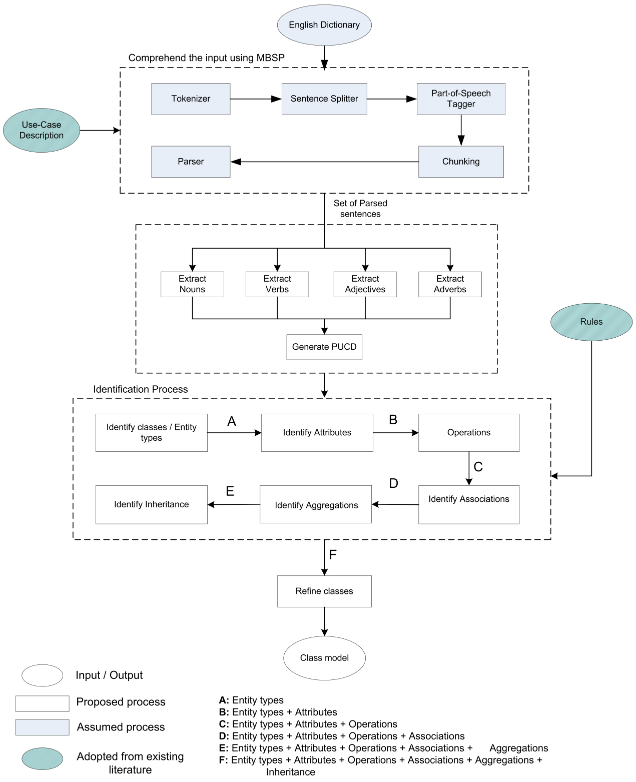


Fig. 2: *Overview of Identification of classes and relationships*

6.2 Comprehension of the Input using MBSP

MBSP is an essential component in text analysis systems for text mining applications such as information extraction and question answering. See [22]. Shallow parsing gives only a partial analysis of the syntactic structure of sentences as opposed to full-sentence parsing. The parsing includes detecting the main constituents of sentences (for example noun phrases (NPs) and verb phrases (VPs)). The MBSP for English consists of the following modules:

(a) Tokenizing: The tokenizer splits a plain text file into tokens. This includes, e.g., separating words and punctuation, identifying numbers, and so on.

(b) Sentence Splitting: The sentence splitter identifies sentence boundaries.

(c) Part-of-Speech (POS) Tagging: The POS tagger assigns to each word in an input sentence its proper part of speech such as noun, verb and determiner to reflect the words syntactic category. See [5, 4].

(d) Chunker: The chunking involves the process of detecting the boundaries between phrases (for example noun phrases) in sentences. See [6]. Chunking can be regarded as light parsing. In MBSP, NL chunking and bracket prediction is applied for the chunking purposes.

(e) Parsing: The parsing here means the process of determining the syntactic structure of a sentence given a formal description of the allowed structures in the language called a *Grammar*.

6.3 Parsed Use-Case Description (PUCD)

The inputs for the PUCD are parsed and tagged text. The main purpose of PUCD is to extract nouns, verbs, adjective and adverbs so as to collect the class/entity type, attribute and relationship from the tagged input.

In some cases the use of one use case description may not be enough to provide all of the information that we need. Therefore, it is recommended to employ more than one use case description to cover all the information needed on properties such as attributes and relationships to produce a class model.

The parsed and tagged text PUCD is defined as a set of tuples as follows:

PUCD = {< original sentence, parsed sentence, Ns, Vs, ADJs, ADVs >}.

Ns = {< N, tag>}, where N is any noun and tag ∈ {NN, NNP, NNPS, NNS}

Vs = {< V, tag>}, where V is any verb and tag ∈ {VB, VBD, VBG, VBN, VBZ}

ADJs = {< ADJ, tag>}, where ADJ is any adjective and tag ∈ {JJ, JJR, JJS}

ADVs = {< ADV, tag>}, where ADV is any adverb and tag ∈ {RB, RBR, RBS}

OS is an original sentence.

PS is a parsed sentence

6.4 Object Identification Process

Details of the object identification process are given in Figure 2. After we extract nouns, verbs, adjectives and adverbs from the generated PUCD, we can then identify classes/entities, attributes and relationships using identification process rules.

Below we illustrate how classes, attributes and relationships are identified:

6.4.1 Identifying Classes/Entities

Figure 2 shows the identification process for identifying classes/entity types, attributes, operations and relationships. We describe more details for each one in the identification process.

The first step in identifying classes is to produce a list of candidate classes. Using PUCD and applying the rules, this can be done by considering all the basic nouns in the PUCD.

A class is defined as follows:

$$C := \{< C_n, \mathrm{ATT}, B, R >\}$$

where C_n is a class name, ATT is a set of attributes, B is a set of behaviour or operations and R is a set of relationships.

We identify a list of candidate classes and attributes as follows:

1. Determiners (such as: a, an, the, each, and, with, etc.) do not play a crucial role at this stage of identification, so they are ignored.

2. Plural noun phrases are converted to their singular form because class names in UML are given in the singular. For example, *customers* is changed to *customer*, and *order items* to *order item.*

3. Redundant candidates are removed from the list of the output PUCD as they are not needed. An exact string matching technique can be used to compare the candidates in the list with each other. For example, customer in our example sentences is a redundant customer which appears many times in the text. The same is done for all other candidates.

Example 1: Identifying objects/classes and relationship with use case description as input.

This example is a very simple one to demonstrate the technique. The inputs for the Parsed Use-Case Description (PUCD) are parsed and tagged text. The main purpose of PUCD is to extract nouns, verbs, adjectives and adverbs that indicate the class/entity type, attribute and relationship from the tagged input. In this example we show the effect of PUCD on one original sentence. This example is simple but we also used many other use case descriptions of varying complexity as input to show the identification of objects/classes and relationships in the production of a class model.

PUCD = {{< OS: Some customers will search for specific CDs or CDs by specific artists, while other customers will want to browse for interesting CDs in certain categories (e.g., rock, jazz, classical), PS:(TOP (S (NP (NNS Customers)) (VP (MD will) (NP (NN access)) (NP (NP (DT the) (NNP Internet) (NNS sales) (NN system)) (SBAR (S (VP (TO to) (VP (VB look) (PP (IN for) (NP (NP (NNS CDs)) (PP (IN of) (NP (NN interest))))))))))) (. .))), Ns:{< NNS Customers >, < NN Access >, < NNP Internet >, < NNS Sales >, < NN System >, < NN Interest>}, Vs: {< Look >} >,

6.4.2 Identify Attributes

A class A has a set of attributes $ATTs$ that describe information about each object:

$$ATT := \{A | A := < A_n, T >\}$$

where each attribute A has an attribute name A_n and a type T.

The first step in the attribute identification process is to extract ADJ and ADV written in the PUCD and apply the attribute rules; this can be done by considering all the basic adjectives and adverbs in the PUCD.

6.4.3 Identifying Relationships

There are four basic kinds of relationships: Association, Aggregation, Composition and Inheritance. Each class C has a set of relationships R. Relationship is represented by relationship type, related class and cardinality. A relationship R is defined as follows:

$R := \{\text{rel} | \text{rel} := < \text{RelType}, \text{relC}, \text{Cr} >\}$

where RelType is a relationship type (e.g., associated with, aggregation, composition and Inherits), relC is a related class and Cr is a cardinality.

7 Result of Building an Initial Class Model using semi-automatic Means

Figure 3 shows a class diagram of the CD Selections Internet System (Place Order Use-Case View) written as functional requirements into the UCDs. This model shows 17 classes drawn as solid rectangles. These classes are linked to each other with associations represented by lines between the class boxes.

We review the production of the refined list of candidate classes and attributes and a list of candidate relationships. Figure 3 shows the result of building an Initial Class Model using semi-automatic means from UCDs as input. The original sentence goes through certain stages: parsed by Memory-based Shallow Parser (MBSP) into tokens, split into sentences, tagged with part-of-speech flag, and identification of noun phrases, verb phrases, adjective phrases and adverb phrases. The next step shows how to extract nouns, verbs, adjectives and adverbs by applying the rules we have identified for classes, attributes, operations and relationships. The candidate classes identified are *Customer, Search Request, CD, CD List, Review.* Three different types of search requests were revealed: *Title Search, Artist Search, Category Search.* By applying the rules to the brief description an additional candidate class identified was *Order.* By reviewing the verbs, contained in this use case, we saw that a *Customer* places an *Order* and that a *Customer* makes a *Search Request.*

Using verbs for identifying relationships is not always straight-forward: a verb may indicate an association or an aggregation or inheritance. The use of NLP to distinguish between these types of relationship is a non-trivial problem, which still needs to be addressed but it is hoped that the use of the whole structure of the PUCD will provide an advance in this area.

We identified a set of attributes for the *Customer* (name, address, e-mail and credit card) and for the *Order* (CDs to purchase and quantity) classes and uncovered additional candidate classes *CD Categories* and *Credit Card Center.* Finally, we realized that the *Category Search* class used the *CD Categories* class, and also identified three subclasses of *CD Categories*, namely *Rock, Jazz, Classical.*

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. Nick Rossiter and Dr. Paul Vickers for agreeing to supervise my research and providing much of his valuable time to that, for all the good discussions. Many thanks to the Cultural Affairs Department, Libyan People's Bureau, Paris, for the help and financial support.

I would be grateful to Dr. Akhtar Ali for his help in the initial formulation of the work described here.

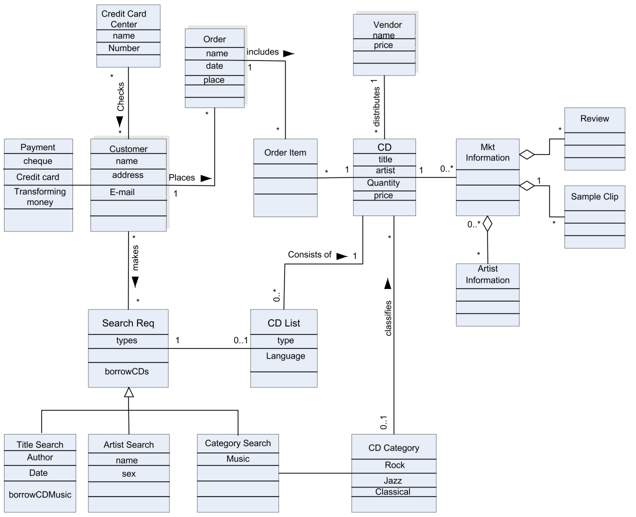


Fig. 3: *Class Model from Use Case Descriptions*

8 Conclusion and Further Work

In this paper we outline an approach that we believe may help to strengthen the process of object identification. We have developed a method called Parse Use-Case Descriptions (PUCDs) to extract nouns, verbs, adjectives and adverbs from use case description. This model is then used for the identification of objects/classes, their attributes, and the static relationships among them to produce a class model. We presented a refinement that generates the class model by using a human expert.

In further work we will focus in with more realistic examples on developing the full method for automatically identifying objects/classes and relationships from RS. We will investigate available technologies and tools to be used, design a system architecture and implement a prototype to realise the identification of entities, attributes and relationships. We will also assess the differences between manually and automatically identifying classes and relationships and evaluate the prototype both in its own right and in a comparison with existing work.

References

[1] R. J. Abbott. Program design by informal English descriptions. *Commun. ACM*, 26(11):882–894, 1983.

[2] B. W. Boehm. *Software Engineering Economics (Prentice-Hall Advances in Computing Science and Technology Series)*. Prentice Hall PTR, October 1983.

[3] G. Booch, J. Rumbaugh, and I. Jacobson. *Object-oriented Analysis and Design With Application*. Addison-Wesley Longman Publishing, Inc., United States of America, 1991.

[4] E. Brill. A simple rule-based part of speech tagger. In *ANLP*, pages 152–155, 1992.

[5] E. Brill. Some advances in transformation-based part of speech tagging. In *AAAI*, pages 722–727, 1994.

[6] W. Daelemans, A. van den Bosch, J. Zavrel, J. Veenstra, S. Buchholz, and B. Busser. Rapid development of nlp modules with memory-based learning. In *In Proceedings of ELSNET in Wonderland*, pages 105–113, 1998.

[7] M. Farid. From English to Formal Specifications, Department of Mathematics and Computer Science. 2000.

[8] F. Gomez, C. Segami, and C. Delaune. A System for the Semi-automatic Generation of E-R Models from Natural Language Specifications. *Data Knowl. Eng.*, 29(1):57–81, 1999.

[9] H. M. Harmain and R. Gaizauskas. Cm-builder: A natural language-based case tool for object-oriented analysis. *Automated Software Engg.*, 10(2):157–181, 2003.

[10] L. Hirschman. The evolution of evaluation: Lessons from the message understanding conferences. *Computer Speech and Language*, 12(4):281–305, 1998.

[11] A. M. Langer. *Analysis and Design of Information Systems*. SpringerLink, United Kingdom, London, 2008.

[12] Y. Liang, D. West, and F. A. Stowel. An approach to object identification, selection and specification in object-oriented analysis. *Inf. Syst. J.*, 8(2):163–, 1998.

[13] D. Liu, K. Subramaniam, A. Eberlein, and B. H. Far. Natural language requirements analysis and class model generation using ucda. In *IEA/AIE'2004: Proceedings of the 17th international conference on Innovations in applied artificial intelligence*, pages 295–304. Springer Springer Verlag Inc, 2004.

[14] B. Macias and S. G. Pulman. A method for controlling the production of specifications in natural language. *Comput. J.*, 38(4):310–318, 1995.

[15] F. Meziane and S. Vadera. Obtaining e-r diagrams semi-automatically from natural language specifications. In *ICEIS (1)*, pages 638–642, 2004.

[16] L. Mich. Nl-oops: from natural language to object oriented requirements using the natural language processing system lolita. *Nat. Lang. Eng.*, 2(2):161–187, 1996.

[17] L. Mich and R. Garigliano. A linguistic approach to the development of object oriented systems using the nl system lolita. In *ISOOMS '94: Proceedings of the International Symposium on Object-Oriented Methodologies and Systems*, pages 371–386, London, UK, 1994. Springer-Verlag.

[18] R. J. Pooley, R. G. Dewar, and K. Li. Object-oriented analysis using natural language processing. 2007.

[19] A. M. Tjoa and L. Berger. Transformation of Requirement Specifications Expressed in Natural Language into an EER Model. In *Proceedings of the 12th International Conference on the Entity-Relationship Approach*, pages 206–217, London, UK, 1994. Springer-Verlag.

[20] R. S. Wahono and B. H. Far. Hybrid Reasoning Architecture for Solving the Object Classes Identification's Problems in the OOExpert System. In *Proceedings of the Annual Conference of JSAI*, pages 351–360, Washington, DC, USA, 2001. IEEE Computer Society.

[21] R. S. Wahono and B. H. Far. OOExpert: Distributed Expert System for Automatic Object-oriented Software Design. In *ICCI '02: Proceedings of the 13th Annual Conference of Japanese Society fo Artificial*, pages 456–457, Tokyo, Japan, 2002. Computer Society.

[22] J. Zavrel and W. Daelemans. Feature-rich memory-based classification for shallow nlp and information extraction. In *Text Mining*, pages 33–54. 2003.

Improving the Output from Software that Generates Multiple Choice Question (MCQ) Test Items Automatically using Controlled Rhetorical Structure Theory

Robert Michael Foster,
University of Wolverhampton,
Wulfruna Street, Wolverhampton, WV1 1LY,
Research Institute in Information and Language Processing
R.M.Foster@wlv.ac.uk

Abstract

A combination of established theories [1].[2].[3].[4] are applied in an attempt to improve the output from a system [5],[6] which automatically generates MCQ (Multiple Choice Question) test items from source documents. The literature observes that NLG (Natural Language Generation) system evaluation is non-trivial [7] and so the method is evaluated using a process suited to the featured domain [8]. The experiment intersperses 38 MCQ test items whose question stems have been generated using Controlled Rhetorical Structure Theory (CRST) with 62 manually created MCQ test items to form an item bank. A usability score is assigned to each item by a domain expert and these scores are used in the evaluation of the effectiveness of the method. The results provide some evidence to support the incorporation of CRST into future versions of the software.

.

Keywords

Controlled Language, Natural Language Generation (NLG), Rhetorical Structure Theory (RST), Multiple Choice Question (MCQ) test item generation, Controlled Rhetorical Structure Theory (CRST)

1. Introduction

Multiple Choice Question (MCQ) test items have been used by the UK Company featured in this study to regularly confirm staff knowledge of documents from the company's Policy Library. The MCQ test items are delivered in the form of pre and post tests associated with training courses and field audits. The stored responses from these tests allow the company to demonstrate that training has been received by staff in accordance with requirements stated in UK Legislation [8]. However an internal study proved that creating and updating the item bank manually is an expensive process. In response to these results we are investigating various ways to automatically generate MCQ test items, the most promising one being the application of a MCQ test item generator [5], [6]. The creators of this system were the only researchers in the field who expressed an interest in collaborating with us in order to improve their system.

The MCQ test item generator [5], [6] uses the following steps to generate MCQ test items:

1. Identify significant terms within the source document
2. Apply a clause filtering module
3. Transform the filtered clauses into questions and
4. Use semantic similarity to select distractors to the correct answer.

During initial experiments with a particular policy document, most of its clauses were filtered out and so the number of usable MCQ test items produced was very small. In order to improve upon this performance various experiments are planned in which a variety of theories about language and learning are applied in pre-processing the source documents. These pre-processing methodologies aim to avoid important clauses being filtered out in step 2 above.

In order to examine the benefits and problems that arise from applying each combination of theories in the pre-processing methodology, a domain specific evaluative measure is used. In the experiment presented in this paper the evaluation is done by analyzing the selections made by a domain expert from a bank of MCQ test items. The relative proportion of automatically generated to manually created MCQ test stems in the selections made by the domain expert is used as an evaluative measure of the proposed adaptations of the system.

The rest of this paper is organized as follows: section 2 describes the motivation for the study and provides a description of Controlled Rhetorical Structure Theory (CRST). Section 3 provides an example to illustrate the application of CRST from source document to output MCQ test item stem and Section 4 describes how the experiment was conducted before presenting the table of results. Conclusions and descriptions of proposed applications can be found in Section 5.

2. Context

2.1 Motivation

The Policy Library for the featured UK Company consists of a small number of general policy statements (POLs) and a large number of Standard Techniques (STs). The STs are intended to give precise instructions for the correct methods that the staff must apply when they are carrying out work on behalf of the company. Several of the Standard Techniques contain requirements for staff to complete sequences of MCQ test items (called 'CBT tests').

For example:

ST:OS7D – Relating to Audits of Operational field staff

"3.1 All Senior Authorised and Authorised Persons who hold an authorisation for HV Operational Work (11SW, 33SW, 66SW, 132SW and restricted variations) shall complete an annual CBT test to the satisfaction of an Examining Officer qualified to examine for that authorisation."

ST:HS17A – The Management of Asbestos Found in Operational, Non Operational Buildings and Equipment

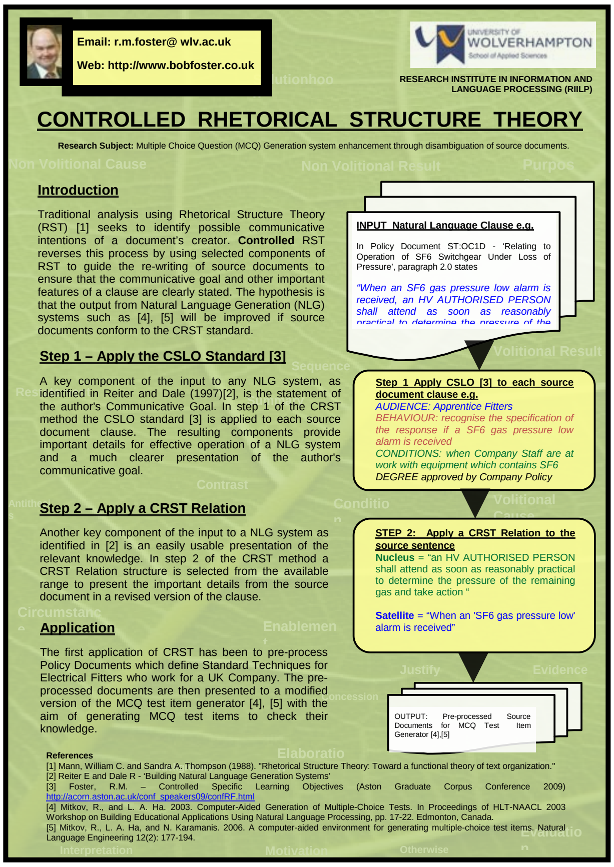
"3.2 After completing the above awareness course all staff shall, on an annual basis, complete the CBT Asbestos Knowledge Refresher which can be accessed via the Safety and Training Resources Catalogue."

In 2007 a review was carried out of the costs of producing and maintaining an item bank of 130 MCQ test items that were first created in 1991. The study demonstrated that item production and maintenance is particularly time consuming. A follow up research project has therefore been set up to analyse and improve the process for creating and maintaining the MCQ assessment tests used by this company.

The most promising approach identified so far has been the use of a MCQ test item generator [5], [6] to generate MCQ test items and post edit them to form the item bank. It has been reported in [5] and [6] that generating MCQ items using the generator can speed up the process by 4 times without compromising the quality of the output. However preliminary experiments applying the system [5], [6] to the policy library from the featured company delivered no usable MCQ test items so significant improvements in performance are necessary before the system could be adopted. This paper describes one attempt to improve system performance by incorporating CRST into both the pre-processing of source documents and the generation of the MCQ test items. An effective method for evaluating the output from this method must be identified and used consistently if the automatic generation of MCQ test items is going to be accepted. The evaluation method chosen must also demonstrate that the generated MCQ test items are as close to manually created ones as possible.

2.2 Controlled Rhetorical Structure Theory

Rhetorical Structure Theory [1] (RST) defines some widely used tools for Natural Language Discourse Processing. Controlled RST (CRST) adapts some of these tools to guide the controlled construction of discourse elements within a well specified domain.



Email: r.m.foster@ wlv.ac.uk

Web: http://www.bobfoster.co.uk

UNIVERSITY OF
WOLVERHAMPTON
School of Applied Sciences

RESEARCH INSTITUTE IN INFORMATION AND LANGUAGE PROCESSING (RIILP)

CONTROLLED RHETORICAL STRUCTURE THEORY

Research Subject: Multiple Choice Question (MCQ) Generation system enhancement through disambiguation of source documents.

Introduction

Traditional analysis using Rhetorical Structure Theory (RST) [1] seeks to identify possible communicative intentions of a document's creator. **Controlled** RST reverses this process by using selected components of RST to guide the re-writing of source documents to ensure that the communicative goal and other important features of a clause are clearly stated. The hypothesis is that the output from Natural Language Generation (NLG) systems such as [4], [5] will be improved if source documents conform to the CRST standard.

INPUT Natural Language Clause e.g.

In Policy Document ST:OC1D - 'Relating to Operation of SF6 Switchgear Under Loss of Pressure', paragraph 2.0 states

"When an SF6 gas pressure low alarm is received, an HV AUTHORISED PERSON shall attend as soon as reasonably practical to determine the pressure of the

Step 1 – Apply the CSLO Standard [3]

A key component of the input to any NLG system, as identified in Reiter and Dale (1997)[2], is the statement of the author's Communicative Goal. In step 1 of the CRST method the CSLO standard [3] is applied to each source document clause. The resulting components provide important details for effective operation of a NLG system and a much clearer presentation of the author's communicative goal.

Step 1 Apply CSLO [3] to each source document clause e.g.

AUDIENCE: Apprentice Fitters

BEHAVIOUR: recognise the specification of the response if a SF6 gas pressure low alarm is received

CONDITIONS: when Company Staff are at work with equipment which contains SF6

DEGREE approved by Company Policy

Step 2 – Apply a CRST Relation

Another key component of the input to a NLG system as identified in [2] is an easily usable presentation of the relevant knowledge. In step 2 of the CRST method a CRST Relation structure is selected from the available range to present the important details from the source document in a revised version of the clause.

STEP 2: Apply a CRST Relation to the source sentence

Nucleus = "an HV AUTHORISED PERSON shall attend as soon as reasonably practical to determine the pressure of the remaining gas and take action "

Satellite = "When an 'SF6 gas pressure low' alarm is received"

Application

The first application of CRST has been to pre-process Policy Documents which define Standard Techniques for Electrical Fitters who work for a UK Company. The pre-processed documents are then presented to a modified version of the MCQ test item generator [4], [5] with the aim of generating MCQ test items to check their knowledge.

OUTPUT: Pre-processed Source Documents for MCQ Test Item Generator [4],[5]

References

[1] Mann, William C. and Sandra A. Thompson (1988). "Rhetorical Structure Theory: Toward a functional theory of text organization."

[2] Reiter E and Dale R - 'Building Natural Language Generation Systems'

[3] Foster, R.M. – Controlled Specific Learning Objectives (Aston Graduate Corpus Conference 2009) http://acorn.aston.ac.uk/conf_speakers09/confRF.html

[4] Mitkov, R., and L. A. Ha. 2003. Computer-Aided Generation of Multiple-Choice Tests. In Proceedings of HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing, pp. 17-22. Edmonton, Canada.

[5] Mitkov, R., L. A. Ha, and N. Karamanis. 2006. A computer-aided environment for generating multiple-choice test items. Natural Language Engineering 12(2): 177-194.

The poster presentation at RANLP 2009 (provided above) explains how CRST unites standard Rhetorical Structure Theory [1] with the theory of Controlled Specific Learning Objectives (CSLO) [4] which in turn incorporates concepts from AECMA Simplified English [3]. The inherent restriction of these theories to well defined domains does not present a problem in the context of this research since the domain is well defined by the company's policy document library from which the source documents are taken. MCQ test item stem templates are applied to the output from the CRST pre-processing and this paper presents the results when the MCQ test items produced are reviewed by a domain expert.

The first step in applying CRST to the process of creating a MCQ item routine is to use established text analysis methods [2].[3] encapsulated within the CSLO standard [4] to produce an unambiguous statement of the communicative goal of the MCQ test item routine. We use the term ‘communicative goal’ in the sense defined in the NLG methodology presented in [11]

The second step is to use these statements of objectives to select an appropriate sequence of CRST templates that when populated will produce an unambiguous presentation of the facts contained within the source document. In the third step of this application of CRST, a series of MCQ test item templates is populated using the content of the CRST-compliant statements generated in step 2.

The hypothesis of the CRST standard is that the translation of source documents into a sequence of CRST templates provides a presentation of the required facts that can be more easily interpreted. The reader of a CRST-compliant document can identify the writer’s communicative goal through the structure of the document i.e. the choices of CRST templates. Also the content words used within the CRST templates are either single sense words, as defined in the AECMA simplified English lexicon [3], or they are words used as defined in a domain specific lexicon compiled for the featured domain.

Disambiguation of this kind is particularly relevant to automatic MCQ test item generation from source documents because it has been noted that the current generator [5], [6] sometimes picks ‘the wrong clause’ during question stem production. The application of the CRST standard enforces clarity of content and the communicative goals of the source document writer and thereby makes such a mistake less likely.

3. Methodology

The examples included in this paper have been chosen to illustrate features of both the proposed source document pre-processing method and the MCQ item stem generation process. They have therefore been taken from the set of generated MCQ test items. However the decision about which items to include as examples was made after the day of the experiment and so their inclusion in no way affected the domain expert’s selections of item stems during the experiment.

CRST is a new application of Rhetorical Structure Theory [1]. This experiment applies CRST both in the source document pre-processing stage and the subsequent MCQ test item generation process. The application of the theory is achieved within a simulation as opposed to a reprogramming of the question generator in order to ensure careful and thorough application of the theory.

The source documents used in the experiment are taken from the policy library of the UK Company that was referred to in the introduction. A description of the method’s application to a particular source document paragraph is presented below in order to illustrate the process that produced the generated stems used in the experiment.

Example 1

In Policy Document ST:OC1D - ‘Relating to Operation of SF6 Switchgear Under Loss of Pressure’, paragraph 2.0 states

“When an SF6 gas pressure low alarm is received, an HV AUTHORISED PERSON shall attend as soon as reasonably practical to determine the pressure of the remaining gas and take action.”

The first step in applying CRST to this paragraph is to define the ‘communicative goal’ of the creator of the MCQ test item routine by applying the CSLO standard [4]. The first draft of this statement in relation to the given paragraph was written as follows:

“Apprentice Fitters recognise the correct description of the response required by Company Policy if a SF6 gas pressure low alarm is received when working with equipment containing SF6”

To ensure this conforms to the CSLO standard, the text is broken up into fragments specifying the Audience, Behaviour, Context and Degree and then re-written to ensure compliance with the controlled Lexicon. This led to several changes but also re-affirmed some of the word choices originally made. This was particularly noticeable within the statement of the required Behaviour where the verb ‘recognise’ was used. This is one of the verbs specified in Bloom’s Taxonomy and is therefore included in the CSLO Controlled Lexicon.

AUDIENCE: Apprentice Fitters

BEHAVIOUR: recognise the specification of the response if a SF6 gas pressure low alarm is received

CONDITIONS: when Company Staff are at work with equipment which contains SF6

DEGREE approved by Company Policy

The full analysis is provided in the table below:

Apprentice Fitters	Domain Specific Lexicon
recognise	CSLO Specific Lexicon
the	Standard Lexicon
specification (changed)	Standard Lexicon
of the	Standard Lexicon
response	CSLO Specific Lexicon
approved (changed)	Standard Lexicon
by	Standard Lexicon
Company Policy	Domain Specific Lexicon
If a	Standard Lexicon

SF6 gas pressure low alarm	Domain Specific Lexicon
Is received when	Standard Lexicon
Company staff (added)	Domain Specific Lexicon
are (added)	Standard Lexicon
at (added)	Standard Lexicon
work (changed)	Domain Specific Lexicon
with equipment	Standard Lexicon
which (added)	Standard Lexicon
contains (changed)	Standard Lexicon
SF6	Domain Specific Lexicon

Table 1 – Analysis and changes made to ensure conformity with the CSLO standard.

This statement clarifies the communicative goal of the creator of the MCQ test item routine, and allows more accurately targeted choices of MCQ test item templates to be made. The objective has been restricted to the checking of apprentices' ability to recognise (level 1 in the Cognitive Domain of Bloom's Taxonomy [9]) the significant facts that they would have been taught during their training. There is no expectation that successful completion of the MCQ test items would provide confirmation of abilities at the other levels within the cognitive domain such as understanding, application, analysis, synthesis or evaluation. Any of these levels which are relevant to the aims of the training and assessment scheme overall are addressed using other approaches, including observation of apprentice work by trainers during training courses and a series of on-site assessments (OSAs).

Having addressed the first step in the application of the CRST method, next comes the choice of an appropriate sequence of CRST templates in order to encapsulate the significant facts. In this case, as was true for many of the sentences processed in this experiment, a single CRST template is sufficient to deliver this content whilst ensuring unambiguous presentation of the communicative goal of the document writer.

The choice of the <Volitional Cause> CRST template in this case was triggered by the presence of the word 'When' in the source sentence:

> **VC Nucleus** = "an HV AUTHORISED PERSON shall attend as soon as reasonably practical to determine the pressure of the remaining gas and take action "
>
> **VC Satellite** = "When an 'SF6 gas pressure low' alarm is received"

This allowed the third and final step whereby one of the MCQ test item templates that can accept the fields defined in the VC CRST template could be applied. In this case the following MCQ test item template was selected:

> "<VC Satellite> What is the first step required by Policy?"

The resulting Generated stem was as follows:

> ***Generated Stem:*** *When an SF6 gas pressure low alarm is received, what is the first step required by Policy?*

This was accepted by the domain expert for inclusion in his MCQ test item routine without any post-processing

> ***Approved Stem:*** *When an SF6 gas pressure low alarm is received, what is the first step required by Policy?*

There now follows a description of how the experiment was conducted and the results of the selection by the Domain Expert.

4. Experiment and Results

On the day of the experiment a full set of 100 MCQ test items was presented to an expert in the featured domain [8]. Before the decision was taken to apply a logical, repeatable method when generating new MCQ items, 62 items had already been manually created using traditional methods. The specification of the job required a set of 100 items to be presented to the subject expert and so only 38 items were required from the application of Controlled Rhetorical Structure Theory (CRST) as described in section 2. Time constraints within the commercial environment prevented the production of any more items.

The general aim of the domain expert in making selections from the bank of 100 items was to confirm apprentices' ability to recognise and recall facts presented during training following their attendance at a series of training sessions. He had no involvement in the creation of either the manually or automatically generated items and had no prior knowledge of which MCQ item stems had been generated. Therefore these factors could not have any bearing upon his decision about which items to include in his MCQ test item routine.

The following usability scores were used to record the domain expert's assessments of the items:

A= Use the item stem unchanged

B= Make minor changes and then use the item

C= Do not use the item

The items used in this experiment have three or four options and there is only one correct answer for each item in accordance with the recommendations from Haladyna and Downing [10]. For this experiment, only domain expert decisions about question stems are reported. Requirements about changes to the options (correct answer and distracters) within the MCQ test item are not reported.

In the case of the MCQ test item example presented in section 3, the item stem was judged by the domain expert to be accurately targeted towards the stated objectives without requiring post processing and so it was placed in category A. Other generated stems required minor post-processing before the domain expert was prepared to use them and still others were categorized ‘not to be used’.

Examples 2 and 3 provide examples of category B decisions. Example 4 is another example of a Category A decision. All generated items stems that were placed in Category C are unsuitable for inclusion in this paper:

Example 2

In Policy Document: ST:SP2A - Relating to Routine Substation Inspection, paragraph 2.1 states that

> *“A substation inspection is a careful scrutiny without dismantling and is normally done with plant and equipment live.”*
>
> **Generated Stem:** What is the definition of 'a substation inspection'?
>
> **Accepted Stem:** What is the Company Policy definition of 'a substation inspection'?

Example 3

In Policy Document ST:SP2J - Relating to The Routine Maintenance of 11kV and 6.6kV Cable Connected Secondary Switchgear (RMU’s, Switches and Fuse Switches), paragraph 2.7 states

> “*Before lowering tanks or working on equipment, ensure mechanism springs are discharged and all trip, close and power supplies are isolated.”*
>
> **Generated Stem:** *When maintaining 11kv Plant in a substation what must Company staff* ***ensure*** *before lowering tanks or working on equipment?*
>
> **Approved: Stem:** *When maintaining 11kv Plant in a substation what must Company staff* ***check*** *before lowering tanks or working on equipment?*

Example 4

In Policy Document ST: HS7C - 'Relating to Rescue from Height Techniques and Procedures, paragraph 3.6 states

> *“Before any rescue from height is attempted, an electrical risk assessment shall be carried out to determine the proximity of any adjacent live circuit.”*
>
> **Generated Stem:** What must be done before any rescue from height is attempted?
>
> **Approved Stem:** What must be done before any rescue from height is attempted?

Once the usability categories were assigned for each of the 100 items in the item bank, the following comparison table was produced:

	Generated MCQ stems	**Manually Created MCQ stems**
A=Use the item stem unchanged	**44% (17 stems)**	**43.5% (27 stems)**
B=Make minor changes and then use the item stem	**31% (12 stems)**	**43.5% (27 stems)**
C= Do not use this item stem	**23% (9 stems)**	**13% (8 stems)**

Table 2 - Usability categorization decisions for Generated vs Manually created MCQ test item stems.

5. Conclusions and Future Work

Applying the decision about category according to a clearly observable action allows the same process to be repeated consistently in other MCQ test item generation experiment evaluations. The separation of acceptable item stems between categories A and B also allows the calculation of three ‘alternative’ evaluation measures (A, B and A and B) as performance improves.

The most encouraging outcome from this experiment is the similarity between the proportions of generated to manually created MCQ test items in both categories A and C. This meets the criteria specified in the introduction whereby generated items need to be indistinguishable from manually created items when viewed by a domain expert. The fact that a slightly lower number of generated items required changes compared to the manually created items might have been due to stylistic differences between the domain expert and the writer of the manually created items.

The most serious limitation with this experiment is that other factors apart from the construction of the item stem will have contributed to the category allocation decision for each item that was made by the domain expert. In fact, the reason that none of the items given category C among the items with generated stems could be provided as examples in section 4 of this paper is because they had been categorized ‘not to be used’ for reasons other than poor construction of the stem. However, in defending the applicability of this experiment I return to the motivation laid out in section 2.1 in which it was clearly stated that generated stems must be indistinguishable from manually created items. This experiment provides very strong evidence to support this hypothesis.

The development effort likely to be involved in creating the software to implement the CRST method is not insignificant. However for the featured domain, the task is feasible because the domain is sufficiently well defined by the policy document corpus which has clearly defined boundaries and is protected by a well organized change management system.

This paper has prepared the ground for future experiments seeking improvement in performance of the featured software [5], [6] by pre-processing source documents. CRST has been applied and a pragmatic, domain specific evaluation method of output from the system has been used. It would not be unreasonable to compare the percentage results for categories A B and C arising from this experiment with similarly gathered results for item banks created using other MCQ test item generation methods.

In future work, other relevant theories of human learning, controlled language and cognitive linguistics will be applied within modified versions of the pre processing methodology as we continue to seek to improve the quality of the output from the MCQ test item generator software [5], [6] in the domain featured in this paper. The main obstacle to applying the method in other domains is the requirement for an AECMA Simplified English style domain specific dictionary. Perhaps future work from our team might provide some techniques for achieving the compilation of such a dictionary semi-automatically.

The evaluation method described in this paper will be applied and refined in future experiments for this project. The most significant refinement will be the inclusion of comparisons with the output from other MCQ test item generation systems.

6. References

[1] Mann, William C. and Sandra A. Thompson (1988). "Rhetorical Structure Theory: Toward a functional theory of text organization."

[2] Mager, R. (1975). Preparing Instructional Objectives (2nd Edition). Belmont, CA: Lake Publishing Co.

[3] AECMA Simplified English, http://www.aecma.org/Publications/SEnglish/senglish.htm 2003-04-16

[4] Foster, R.M. – Controlled Specific Learning Objectives (Aston Graduate Corpus Conference 2009) http://acorn.aston.ac.uk/conf_speakers09/confRF.html

[5] Mitkov, R., and L. A. Ha. 2003. Computer-Aided Generation of Multiple-Choice Tests. In Proceedings of HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing, pp. 17-22. Edmonton, Canada.

[6] Mitkov, R., L. A. Ha, and N. Karamanis. 2006. A computer-aided environment for generating multiple-choice test items. Natural Language Engineering 12(2): 177-194.

[7] Reiter, E and Belz, A – Title: A Proposal for Shared-task Evaluation in NLG

[8] UK Legislation Health and Safety at work, etc Act 1974 http://www.hse.gov.uk/legislation/hswa.pdf

[9] Bloom, B. (1956), Taxonomy of Educational Objectives: Book 1 Cognitive Domain, Longman, 1956.

[10] Haladyna TM, Downing SM. How many options is enough for a multiple-choice test item? Educ Psychol Meas. 1993;53:999–1009

[11] Reiter E and Dale R - ‘Building Natural Language Generation Systems’

Normalized Accessor Variety Combined with Conditional Random Fields in Chinese Word Segmentation

Saike HE†
Beijing University of Posts and Telecommunications
Beijing, 100876, China
hsk000@gmail.com

Taozheng ZHANG
Beijing University of Posts and Telecommunications
Beijing, 100876, China
zhangtaozheng@gmail.com

Xue BAI
Beijing University of Posts and Telecommunications
Beijing, 100876, China
bc003@sina.com

Xiaojie WANG
Beijing University of Posts and Telecommunications
Beijing, 100876, China
xjwang@bupt.edu.cn

Yuan DONG
France Telecom R&D Center
Beijing, 100080, China
yuandong@orange-ft.com

Abstract

The word is the basic unit in natural language processing (NLP), as it is at the lexical level upon which further processing rests. The lack of word delimiters such as spaces in Chinese texts makes Chinese word segmentation (CWS) an interesting while challenging issue. This paper describes the in-depth research following our participation in the fourth International Chinese Language Processing Bakeoff [1]. Originally, we incorporate unsupervised segmentation into Conditional Random Fields (CRFs) in the purpose of dealing with unknown words. Normalization is delicately involved in order to cater to problem of small data size. Experiments on CWS corpora from Bakeoff-4 present comparable results with state-of-the-art performance.

Keywords

Unsupervised Segmentation, Conditional Random Fields, Normalized Accessor Variety.

1. Introduction

Words are the basic linguistic units of natural language. However, Chinese texts are character based, not word based. Thus, the identification of lexical words or the delimitation of words in running texts is a prerequisite of NLP.

Chinese word segmentation can be cast as simple and effective formulation of character sequence labeling. A prevailing technique for this kind of labeling task would be Conditional Random Fields1 (CRFs) [1], following the current trend of applying machine learning as a core technology in the field of natural language processing. Based on conditional dependency assumption, CRFs could exert predominant performance on the known words (which refer to those words exist in both the testing and training data), yet further improvement for CWS systems are usually limited by the comparative large fraction of unknown words (which refer to those words exist only in the testing data).

Regarding this nontrivial issue, in this paper, we are intended to provide a semi-supervised methodology: incorporates an unsupervised method into supervised segmentation, following the in-depth research after our participation in Bakeoff-4. Catering to the common case of limited training data, normalization is involved in the unsupervised phrase.

The rest of the paper is organized as follows: Section 2 describes the framework of our CWS system in detail. Section 3 discusses the unsupervised segmentation method based on a modified version of the target function. Section 4 presents and analyzes our experimental results. Finally, we conclude the work in Section 5.

2. Framework of CWS

Our framework of CWS utilizes Conditional Random Fields (CRFs) as the basic statistical model. The Tag set and features used to train CRFs are also introduced briefly in this section.

2.1 Conditional random fields

Conditional random fields (CRFs) for sequence labeling offer advantages over both generative models like HMMs and classifiers applied at each sequence position [2]. CRFs are an undirected graph established on $G = (V, E)$, where V is the set of random variables $Y = \{Y_i | 1 \leq i \leq n\}$ for each the n tokens in an input sequence and $E = \{(Y_{i-1}, Y_i) \ |2 \leq i \leq n\}$ is the set of (n − 1) edges forming a linear chain. Following [1], the conditional probability of the state sequence $(s_1, s_2 \ldots s_n)$ given the input sequence $(o_1, o_2 \ldots o_n)$ is computed as follows:

[1] The Fourth International Chinese Language Processing Bakeoff & the First CIPS Chinese Language Processing Evaluation (Bakeoff-4), at: http://www.china-language.gov.cn/bakeoff08/bakeoff-08_basic.html

$$P_{\Lambda}(s \mid o) = \frac{1}{Z_o} \prod_{c \in C(s,o)} \exp\left(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(s_{t-1}, s_t, o, t)\right) \quad (1)$$

where f_k is an arbitrary feature function; and λ_k is the weight for each feature function; it can be optimized through iterative algorithms like GIS [3]. Recent research indicates that quasi-Newton methods such as L-BFGS [4] are more effective than GIS.

2.2 Tag set

As justified in [5, 6], a 6-tag set enables the CRFs learning of character tagging to achieve a better segmentation performance than others. So we adopt this tag set in our CWS framework， namely, B, B2, B3, M, E and S, which respectively indicates the start of a word, the second position within a word, the third position within a word, other positions within a word, and the end of a word. An example is illustrated in Table 1.

Table 1. Illustration of 6-tag format in CWS

Word Length	Tag sequence for a word
1	S
2	BE
3	BB2E
4	BB2B3E
5	BB2B3ME
≥6	BB2B3M ••• ME

2.3 Feature templates

Table 2. The features used in CWS systems.

Type	Feature
Unigram	C_n(n=-2,-1,0,1,2)
Bigram	C_nC_{n+1}(n=-2,-1,0,1)
Jump	$C_{-1}C_1$
Punctuation	Pun(C0)
Date,Digit,letter	T_{-1} T_0T_1

Table 2 illustrates the features we used in our CWS systems. Where *C* represents character; subscript n indicates its relative position taking the current character as its reference; *Pun* derives from the property of the current character: whether it is a punctuation; *T* describes the type of the character: numerical characters belong to class 1, characters whose meanings are date and time represent class 2, English letters represent class 3, punctuation labels represent class 4 while other characters represent class 5. In addition, the tag bi-gram feature is also employed.

3. Unsupervised segmentation

Although CRFs model could tackle the known words accurately based on the information learned from the training data, the segmentation on the unknown words rests on reliable statistical information derived from large amount of running texts. Thus, we resort to unsupervised segmentation method to deal with these unknown words. In general, unsupervised segmentation assumes no label information for training. It rests on statistical information over the whole corpus to identify potential words, each assigned a goodness score to indicate their credibility. In this section we will introduce an existing unsupervised segmentation criterion, whose segmentation results are encoded into additional features to facilitate supervised learning for CWS. To make it more reliable, normalization strategy is involved.

3.1 Accessor variety

In Chinese text, each substring of a whole sentence can potentially form a word, but only some substrings carry clear meanings and thus form a correct word. Accessor variety (AV), sparked by [7] is used to evaluate how independent a string is from the rest of the text. The more independent it is, the higher the possibility that it is a potential word carrying a certain kind of meaning. The accessor variety (AV value) of a string *s* is defined as:

$$AV(s) = \min\{Lav(s), Rav(s)\} \quad (2)$$

where Lav(s) is the left accessor variety of s, which is defined as the number of its distinct predecessors, plus the number of distinct sentences in which s appears at the beginning, while Rav(s) is the right accessor variety of s, which is defined as the number of its distinct successors, plus the number of distinct sentences in which s appears at the end.

3.2 Unsupervised segmentation

Given the formula for calculating the AV value of a certain string within a sentence, the segmentation problem is then cast as an optimization problem to maximize the target function of the AV value over all word candidates in a sentence. For the sake of convenient, we use a *segmentation* to denote a segmented sentence, a *segment* to denote a continuous substring in the segmentation, and *f* to denote the target function. We use *s* to represent a string (e.g. a sentence), *S* to represent a segmentation of *s*, n to represent the number of characters in *s*, and m to denote the number of segmentation in *S*. The sentence *s* can be displayed as the concatenation of n characters, and S as the concatenation of m strings:

$$s = c_1c_2c_3 \ldots c_i \ldots c_n$$

$$S = w_1w_2w_3 \ldots w_i \ldots w_m$$

where c_i stands for a character and w_i stands for a *segment*. The target functions *f* is given below [8]:

$$f(\mathrm{S}) = \sum_{i=1}^{m} f(\mathrm{w}_i) \qquad (3)$$

Given a target function *f* and a particular sentence *s*, we need to choose the *segmentation* that maximizes the values of *f*(S) over all the possible segmentations. In formulation function *f*(w) ,we consider two factors: one is the segment length, denoted as |w|, and the other is the AV value of a segment, denoted as AV(w). Then, f(w) can be formulated as a function of |w| and AV(w), thus the target function can be regarded as a choice of normalization for the AV(w) to balance the segmentation length and the AV value for each segmentation. Theoretically, the choice of *f*(w) is arbitrary, among the most representative types of functions (namely, polynomial, exponential, and logarithmic functions), we choose polynomial function for *f*(w) (hereafter, referred as AV), since it proves to be the best in our CWS system, and it is defined as:

$$f(\mathrm{w}) = |\mathrm{w}|^{c} \times \mathrm{AV}^{d}(\mathrm{w}) \qquad (4)$$

where c and d are integer parameters that are used to define the target function *f*(w), whose performance has been justified in [8].

As the training is usually too limited, then there would be a great chance that fluctuation exists in the AV value of a string consist of extreme number of characters, that is to say: there should be a disparity between dealing with strings with very few characters and that with much more characters when calculating AV values. Such fluctuation may deteriorate the reliability of AV value in that: single-character candidate, such as stop word or interrogative marker, may receive comparatively low AV value, though considering them as an isolate word is actually much better; multi-character potential word, which carries no practical meaning is highly possible to obtain a relatively high AV value just because there is a high concurrence frequency among those characters. Unfortunately, both of these flaws inherent in formula (4) are overlooked in either [6] or [8], at least not mentioned in detail. To deal with this special case, as well as alleviate the fluctuation in AV values, we introduce a normalized version of formulation function *f*$_N$(w) (hereafter, referred as NAV) in accessor variety, as formulated below:

$$f_{\mathrm{N}}(\mathrm{w}) = \frac{|\mathrm{w}|^{c} \times \mathrm{AV}^{d}(\mathrm{w})}{1 + \left(\frac{|\mathrm{w}|}{\mathrm{Norm}}\right)} \qquad (5)$$

A real-value normalizer, named as *Norm* is involved in (4) to obtain (5). The modified formulation function f_N is based on the following consideration: on the one hand, when |w| is large enough, unless its accessor variety is relative high, it would not be considered as a potential word, thereby a low value would be assigned to the current segment strategy; on the other hand, when |w| is too small, unless its accessor variety is also relative low, it would still enjoy high favor, the current segment strategy receives comparably high value accordingly. This measure coincides with that proposed in [8], with a superiority of the absence of special consideration for single character or multi-character candidates.

With all the information above prepared, here comes the computation of *f*(S) for a given sentence *s*. Since the value of each segment can be computed independently from the other segments in *S*, *f*(S) can be computed using a dynamic programming technique, in which the time complexity is linear to sentence length. Let us use $\mathrm{f_i}$ to denote the optimal target function value for the sub-sentence $\mathrm{c_1c_2…c_i}$ and $\mathrm{w_{j…i}}$ to denote the segment $\mathrm{c_{j+1}c_{j+2}…c_i}$ (for j ⩽ i). Then we have the following dynamic equations:

$$\mathrm{f_0 = 0;}$$

$$\mathrm{f_1 = f(w_{1...1} = c1);}$$

$$\mathrm{f_i} = \max_{0<\mathrm{i-j}<7} \mathrm{f_j + f(w_{j...i}),\ for\ i > 1;}$$

$$\mathrm{f}(S) = \mathrm{f_n}.$$

It is worth noticing that in each iteration, there are at most *N*(in our experiment *N = 6*) possible choice, where N is the maximum length of a word.

3.3 AV feature

Having nailed down the definition of accessor variety and target function, we could conduct the unsupervised segmentation. However, we now confront two choices to utilize the AV feature: (1) using the unsupervised segmentation result (in the form of 6-tag set as mentioned in section 2 as auxiliary feature for each character within a sentence s in training CRFs. (hereafter, referred as ‘Auxiliary Seg’) (2) directly assigning the AV value calculated by formula (5) to a string under the best segmentation S for sentence s (hereafter, referred as ‘NAV value’). In the latter case, we need to define a feature function to narrow down the value span of AV feature to avoid the problem of data sparsity. Here, we adopt the same feature function in [6], which is defined as

$$\mathrm{f_n(s) = t,\ if\ 2^t \le AV(s) < 2^{t+1}} \qquad (6)$$

where t is an integer to logarithmize the score.

Without any single piece of proof that either of two methods of utilizing AV feature is superior to the other, controlled experiment is conducted in section 4 to seek for an explicit conclusion to this issue.

4. Evaluation results

This section reports the experiment result based on CWS corpora from Bakeoff-4. The corpora consists of 5 data sets, namely, CITYU, CKIP, CTB, NCC and SXU on both closed and open tracks. The corpus from MSRA is simplified Chinese text while the other corpora are in traditional Chinese. The original label for the training data set is IOB-2. Here, we convert all the corpora to 6-tag set as introduced in section 2.2.

4.1 Subsections experiment setting

In the unsupervised method (both AV and NAV), maximal segment length of potential word is set to 6. The two parameters c, d in formula (4) and (5) are set to 1, and 2 respectively, followed by the best setting achieved in our CWS system. Notice, the calculation of AV values in the phrase of unsupervised segmentation are derived from both training and testing corpus (in unsupervised segmentation, the training data is utilized as unlabeled data as well).

4.2 Two ways of utilizing AV value

To find out the better strategy to utilize accessor variety, we conduct a controlled experiment on the close tracks, that is： CWS with AV, CWS with NAV, and the result is shown in Table 3.

Table 3. Comparison between two ways of utilizing AV value

Run ID	F-Score	
	Auxiliary Seg	NAV value
CITYU	94.50	94.93
CKIP	93.21	94.04
CTB	94.89	95. 39
NCC	92.41	93.93
SXU	95.63	96.19

(Note: the parameter *Norm* in formula (5) for NAV is set to 2.5)

The final result indicates that the strategy with 'NAV value' presents better performance. This may be explained as the error brought in by the 'Auxiliary Seg' which promulgates through the whole sentence thus misguides the CRFs learner.

4.3 'Norm' parameter setting in NAV

Table 4. The result of NAV on CWS closed tracks with different settings of parameter Norm

Run ID	F-Score		
	Norm=2	Norm =2.5	Norm =3
CITYU	94.92	94.93	94.87
CKIP	93.94	94.04	94.05
CTB	95.50	95.39	95.35
NCC	93.91	93.93	93.94
SXU	96.15	96.19	96.08

As we can see from Table 4, NAV achieves comparatively higher performance when Norm is set to 2.5. Our experiment implies that when parameter Norm is set within the span between 2 to 3, relatively performance promotion can be obtained. For the sake of convenience, the parameter *Norm* in formula (5) for NAV is set to 2.5 in the following experiments.

4.4 Performance of four systems

For the purpose of comparison, Table 5 lists the performance of four systems on the close tracks.

Table 5. The results of four systems on CWS closed tracks[2]

Run ID	F-Score			
	baseline	**+AV**	**+NAV**	**best**
CITYU	94.43	94.78	94.93	95.10
CKIP	93.17	93.90	94.04	94.70
CTB	94.86	95. 45	**95.39**	95.89
NCC	92.99	93.00	93.93	94.05
SXU	95.46	96.15	96.19	96.23

Where 'baseline' presents our CWS system participating in Bakeoff-4, which only utilizes the feature defined in Table 2. '+AV' indicates AV features are applied; '+NAV' indicates normalized NAV features are involved; while 'best' indicate the topline achieved in Bakeoff-4. Close scrutiny to Table 5 indicates '+AV' can lift the performance of the original CWS ('baseline') to a comparatively higher position, while '+NAV' performs best and are really comparative to the topline result. For the performance improvement of NAV, the normalization mechanism in formula (5) plays a key role. However, it is necessary to point out that the performance of CTB is slightly drawn down by NAV feature compared to that of AV , yet still higher than the 'baseline' system. The value, 2.5 for Norm may not be a proper setting, which can serve as a reasonable explanation for this abnormal phenomenon.

4.5 Performance of CWS open tracks

In this experiment group, we will report the performance of NAV on the open tracks.

In the open tracks, corpus from previous bakeoffs are involved to train CRFs. Additionally, transformation-based error-driven learning (TBL) is also involved and used in

[2] The evaluation tool can be downloaded from http://www.china-language.gov.cn/bakeoff08/

the post-processing phrase. Table 6 lists the corpora used to train the CRFs and TBL learner in the open tracks.

Table 6. Corpora used to train the CRFs classifier and the TBL learner

Run ID	CRFs	TBL
CityU	2005,2006,2007	2003
CKIP	2007	2006
CTB	2006,2007	2007

This experiment group aims at clarifying whether NAV could bring further performance promotion for CRFs in open tracks. As a great amount of external resource is involved, the space for improvement left for NAV is really limited, thus proves to be a challenging task for NAV. Table 7 lists the result of NAV and four comparison systems on the CWS open tracks.

Table 7. The results of four systems on CWS open tracks

Run ID	**F-Score**			
	baseline	**+AV**	**+NAV**	**best**
CITYU	96.97	97.00	**96.99**	**96.97**
CKIP	93.64	94.48	94.53	95.63
CTB	97.93	97.94	97.96	99.20
NCC	-	-	-	
SXU	-	-	-	

(In this experiment setting, we did not conduct experiments on NCC or SXU since no extra data are available for us on these two data sets.)

With a stronger CRFs model and an additional TBL learner, the performance of 'baseline' system are boosted to a much higher level, as we can see from the comparison of Table 6 and Table 7. Still, performance promotion does occur under such circumstance, and the result brought by NAV (96.99) even surpass the topline (96.97) on CITYU data set. Thus, it demonstrates that accessor variety is also useful in the case of open tracks where large amount of external resource are involved, and Normalized accessor variety turns out to be more effective than original AV value.

5. Conclusions

In this paper, we have proposed an effective method of incorporating unsupervised segmentation method into CRFs model. To make the unsupervised strategy more reliable, normalization strategy is involved. Our experiments justify that accessor variety used as 'NAV value' presents better performance over 'Auxiliary Seg' strategy. Although a core parameter Norm, which if differ in diverse settings, will bring about different results in the final evaluation, creditable performance promotion can be obtained within a certain span. In the closed tracks of Bakoff-4, CRFs model with NAV method achieves comparable performance with the topline; while in the open tracks, NAV is still useful when large amount of external resource are involved. Thus, NAV provides us with a effective way to further boost the performance of Chinese Word Segmentation.

6. References

[1] J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the 18th ICML, 282–289, San Francisco, CA.

[2] F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields. In Proc. of HLT/NAACL-2003, 134-141. Edmonton, Canada.

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

[4] R.H. Byrd, J. Nocedal and R.B. Schnabel. 1994. Representations of quasi-Newton matrices and their use in limited memory methods. Mathematical Programming, (63):129-156.

[5] Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2006b. Effective tag set selection in Chinese word segmentation via conditional random field modeling. In PACLIC-20, pages 87–94, Wuhan, China, November 1-3.

[6] Hai Zhao and Chunyu Kit, 2008. Unsupervised Segmentation Helps Supervised Learning of Character Tagging for Word Segmentation and Named Entity Recognition, The Sixth SIGHAN Workshop on Chinese Language Processing (SIGHAN-6), pp.106-111, Hyderabad, India, January 11-12.

[7] Haodi Feng, Kang Chen, Chunyu Kit, and Xiaotie Deng. 2005. Unsupervised segmentation of Chinese corpus using accessor variety. In K.-Y. Su, J. Tsujii, J. H. Lee, and O. Y. Kwong, editors, Natural Language Processing- IJCNLP 2004, volume 3248 of Lecture Notes in Computer Science, pages 694–703, Sanya, Hainan Island, China. Springer Berlin / Heidelberg.

[8] Haodi Feng, Kang Chen, Chunyu Kit, and Xiaotie Deng. 2005. Unsupervised segmentation of Chinese corpus using accessor variety. In K.-Y. Su, J. Tsujii, J. H. Lee, and O. Y. Kwong, editors, Natural Language Processing - IJCNLP 2004, volume 3248 of Lecture Notes in Computer Science, pages 694–703, Sanya, Hainan Island,China. Springer Berlin / Heidelberg.

Event Ordering. Temporal Annotation on Top of the BulTreeBank corpus

Laska Laskova
Institute for Parallel Processing, Bulgarian Academy of Sciences
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
Department of Bulgarian Language, Sofia University “St. Kl. Ohridski”
15 Tsar Osvoboditel Blvd., 1504 Sofia, Bulgaria
laska@bultreebank.org

Abstract

This paper describes the preliminary work on the project of extending the BulTreeBank with temporal information that will serve as a golden standard for Bulgarian language. We outline a flexible markup scheme that is based on a language-specific verb taxonomy and test its capabilities by implementing algorithms for temporal entities recognition in the CLaRK System tool.

Keywords

temporal expressions, temporal relations annotation, verb categories, boundedness

1. Introduction

Recently, an extensive work is being done on the automatic recognition and normalization of temporal expressions in natural languages (e.g. the MUC 6 and MUC 7 Named Entity Recognition Task, the Temporal Expression Recognition and Normalization Task). We propose a TimeML-based annotation scheme for temporal expressions in Bulgarian. The original scheme [9] was modified so that the annotation could benefit from the language-specific means for conveying temporal information: lexical aspectual type, Slavic Aspect (the so called *vid* category), tense and evidentiality. In Bulgarian, a language with rich verbal morphology, they play a crucial role in temporal order decoding.

Our final aim is to facilitate the creation of a gold standard by annotating automatically some of the temporal information. On structure level we focus on the interaction between verb phrases and temporal function words (conjunctions and prepositions). The technical part is carried out using the BulTreeBank, an HPSG syntactically annotated corpus of Bulgarian [11]. A rule-based algorithm for temporal relations detection is implemented in the XML-based CLaRK System [12]. Its performance proves that morphologically encoded aspectual data is important when analyzing temporal relations for Bulgarian.

2. Exploiting Bulgarian verb categories

Although when analyzing temporal relations (TRs) we would like to take into account world-knowledge information, especially causation and knowledge of language usage, at this stage of annotation we do not have the resources to complete such a task in a short time. We decided to calculate automatically temporal relations, which depend solely on sentential syntax, word order, morphological and limited lexical information. In order to achieve this goal we have systematized the information that can be found in the existing descriptive literature [2]. Our next step on this preliminary stage was to develop a taxonomy of lexical aspectual types, which proved to be relevant for encoding temporal ordering.

2.1 Aspectual verb classification

Verbal aspect category *vid* has two subcategories – namely, imperfective (IPF) and perfective (PF). Verbs are overtly marked for their *vid*, except for a relatively small group of biaspectual verbs in third declension. We accept that for Bulgarian language *vid* category encodes information about the boundedness of the eventuality denoted by the verb. This, of course, does not imply that the aspectual type of the verb is fixed, but we argue that this feature imposes some rigid limitations concerning the scope on the structure of the event, and hence some restrictions on the set of possible aspectual properties of the verb [6]. That is why we have decided to build our verb classification with respect to which nucleus element(s) verbs are related to. The well-known nucleus components (Figure 1) are described in the works of Moens and Steedman [8].

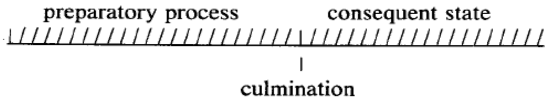


Figure 1. Nucleus structure.

Further subcategorization based on Vendlerian lexical aspectual classification is made with respect to affixation. For Slavic languages like Polish, Bulgarian, Russian and so on it has long been known that the aspectual type is marked by word-formational features and changed through derivational processes (just to mention a few recent studies: [3], [12], [6]). Verb classes whose differences proved to be relevant for TRs recognition are listed below.

2.1.1 Imperfective stem verbs

These are atelic verbs that focus on the unboundedness of the eventuality – states and activities, which are not related to any nucleus as its preparatory process.

2.1.2 Perfective verbs

Here we distinguish three groups. Telic stem verbs are typically achievements or accomplishments (culminated processes in Moens and Steedman terminology). The former focuses only on the culmination of the event structure and the latter both on the preparatory process and the culmination. The same holds for telic verbs derived by prefixes from imperfective base verbs.

Delimitatives derived by *po-* and *nad-* prefixation and expressing bounded but atelic eventualities are accomplishment verbs. In contrast, utterances with the so-called majorative-resultatives, which express activity that ends “beyond the proper limit” [5], e.g. *prejam* – “to have eaten too much”, could equally receive the accomplishment as well as the achievement profile. Both classes focus on a process, but in the first case this process does not belong to a nucleus structure, and in the second it is identified as a preparatory process.

Verbs derived by -n_1- suffixation express punctual events with no internal structure. Only few of them denote points that are not incorporated in a nucleus structure. Most of these verbs could receive ingressive reading, focusing on the point which serves as the initial bound for the process. Either way, we treat all -n_1- perfectives as achievement verbs. This inconsistency is corrected on the level of TR annotation. When the perfective verb has a semelfactive reading, it is marked as *MOMENT*, but for an ingressive reading it receives *INITIATION* markup (see Table 1).

2.1.3 Secondary imperfective verbs

Verbs derived from perfectives by the *-a-* suffix or *-v-* suffix and its variants focus on the preparation process in the nucleus structure. For this reason, in many contexts the realization of the culminated process is implied, especially in a present historical tense, and on a number of occasions the nucleus component referred to by the utterance is not the process, but the culmination itself.

2.1.4 Ingressive and terminative verbs

Bulgarian perfective ingressive verbs, prefixed with *pro-* and *za-*, and terminative verbs, prefixed with *do-*, are derived from their imperfective counterparts: *zapeja* (PF) → *zapjavam* (IPF), “to start singing”, *dopeja* (PF) → *dopjavam* (IPF), “to finish singing”. Since perfectives focus on the process starting point, respectively culmination, they are assigned aspectual class achievement (that can be shifted to accomplishment). Again, for ingressive verbs this is obviously not the most adequate interpretation, but it suits us for the moment. On the other hand, imperfectives are assigned aspectual class activity (that can be shifted to achievement), because they focus on the beginning phase of a process, not necessarily culminated or otherwise limited, respectively the finishing phase of a culminated process that is implied to be interrupted.

2.1.5 Encoding aspectual class

Since on a token level verb forms in the BulTreeBank corpus are annotated with morphosyntactic tags providing information about *vid* category [10], we decided to use yet another attribute, *AspCat* (Aspectual Category). In accordance with the above classification, this tag receives one of the following five values: *state, act, ach, acc-ach, acc-act* (corresponding to Vendlerian types state, activity, achievement, accomplishment or achievement, accomplishment or activity). The ambiguity of the values is intended. The introduced attribute is not part of the tag set for temporal information mark-up. For the moment, the annotation is done manually but is computer-assisted[1]. Verbs that have only iterative readings are regarded as processes and their *AspCat* attribute receives *act* value, but on the level of TRs annotation they are further subcategorized as *SERIES*.

2.1.6 Encoding phase

Bulgarian verbs encode not only information about the type of eventuality expressed, but also about its phase. TimeML temporal annotation scheme provides a special mark-up for aspectual verbs and their complements, but we have to employ another attribute for ingressives and terminatives, namely, *@phase* (Table 1).

3. TimeML adopted for Bulgarian

TimeML emerged as a markup language for time, events and temporal links after the TERQAS workshop held in 2002 [9]. Temporal information should be represented via several tag types: EVENT – for event tokens, where event is any kind of situation that happens or occurs, MAKEINSTANCE for event instances (in contrast to event tokens), SIGNAL for textual elements that explicitly mark temporal or modal relations and quantification over events, TIMEX3 for temporal expressions, and LINK for relationships. The LINK tag is always one of the following types: TLINK (Temporal Link) for relations between two events or an event and a time, SLINK (Subordination Link) for relations between two events or an event and a signal, and ALINK (Aspectual Link) for relations between an aspectual event and its argument event.

The corpus annotated according to TimeML, TimeBank, comprises English newspaper articles marked for temporal information only, but our corpus is HPSG syntactically annotated on HPSG-based grounds, which, besides language specificity, calls for altering some of the TimeML tags.

[1] In the future this task will be accomplished by means of a regular grammar, based on the lexicon of Bulgarian verb Aktionsarten [5], revised accordingly to the classification presented in this paper (section 2.1.).

Table 1. EVENT attributes for Bulgarian

EVENT ATTRIBUTE	*VALUES*	*ACCOUNTS FOR*
aspect, altered	STATE, ACTIVITY, ACHIEVEMENT, ACCOMPLISHMENT, MOMENT, SERIES, NOT_SPECIFIED	Vendlerian aspectual class; note that two more classes are added – *series* for iterative "one episode" eventuality [4], and *point* for semelfactives [8]
class, altered	REPORTING, PERCEPTION, ASPECTUAL, INTENSIONAL, OTHER	lexical meaning. Events of type I_STATE or I_ACTIVITY (Intensional State, resp. Activity) are thus "decomposed" and signaled by two attribute values
conState, new	FACT, RESULT	lack or presence of culmination in the event structure for the related verbs in the present and past perfect tense; achievements, accomplishments *and* points are opposed to states and activities
evid, new	INDICATIVE, RENARRATIVE, CONCLUSIVE, DUBITATIVE	verbal evidentiality feature. It indicates both with the source of information and speaker's attitude about the statement validity
location, new	PAST, PRESENT, FUTURE, NONE	orientation of the event with regard to the perspective time – document creation time or other. BulTreeBank provides tense information derivable from the morphosyntactic tags [10]
persp Anchor, new	boolean	event potential to anchor shift of perspective. Its default value is true for perceptive and reporting verbs
phase, new	INITIATION, TERMINATION	beginning or end phase of the eventuality, encoded morphologically
status, new	NEGATIVE, POSITIVE	lack or presence of verb negation. When negated, verbs are treated as denoting moments or intervals where a particular situation does not hold

3.1 Adjustments of the annotation for the BulTreeBank corpus

All TimeML tags are represented as empty daughter elements with an appropriate attribute set. LINKs for intersentential relationships are embedded under the sentence node, TIMEX3, SIGNAL and EVENT elements are daughters in first position of the relevant lexical or phrasal node.

3.1.1 EVENT element

On this stage we annotate automatically only events expressed by means of verbs. In the BulTreeBank annotation scheme, verb complex, i.e. finite verb, accompanied by clitics, auxiliary particles (auxiliary verb forms and negative particles), participles and emphatic adverbs, is considered as a multi-token verb [10]. For this reason, some of the relations between event and signal, for example, are annotated not by means of LINK, but as a value for EVENT tag attribute.

The EVENT element introduced for the needs of the BulTreeBank temporal annotation is different from its TimeML counterpart. New optional attributes were added, and some of the values of the old attributes were altered. The differences are summarized in Table 1.

3.1.2 Other elements

There are some other changes in the scheme but due to the lack of space we cannot present them here. Since we focus on temporal ordering between events, we have to mention at least two of them. Originally, *RelType* attribute for TLINK has 13 possible values, based on James Allen's [1] 13 interval-interval and interval-moment relations: *BEFORE, AFTER, IBEFORE, IAFTER, INCLUDES, IS_INCLUDED, HOLDS, SIMULTANEOUS, IDENTITY, BEGINS, ENDS, BEGUN_BY, ENDED_BY.*

In our scheme @RelType is required, so we add the 14th value VAGUE, for temporal relations that are ambiguous or cannot by assigned automatically.

SLINK will not represent a relationship between a SIGNAL for negation particles and an EVENT when the verb is negated. Instead, this information will be encoded via the *status* attribute. As a consequence, ELINK (Entailment Link) is introduced to describe entailed TRs between an eventuality and a negation argument situation.

4. Experiment

Our aim was to test a rule-based approach for detecting TRs between events by employing information about sentential syntax, word order, temporal signal, tense, verb negation and sets of possible aspectual types.

The experiment was performed on a small set of 132 two-clause sentences extracted from the BulTreeBank corpus. 118 sentences of them are verbal head-adjunct phrases, 14 – coordinated phrases, in both cases clauses are connected by *dokato* conjunction. As a coordinating conjunction it corresponds to English "whereas". Subordinating *dokato* regarded as ambiguous: the two eventualities could be overlapping ("while"), or, one of the events, regardless of constituents' relation, is ended by the other ("until"). For instance:

(1) *Докато те чаках, гледах телевизия.*

while you.ACC wait.1sg.IPF.IMPERFECT watch.1sg.IPF.IMPERF TV

"**While** waiting for you, I was watching TV",

(2) *Гледах телевизия, докато (не) дойде сестра ти.*

watch.1sg.IPF.IMPERF TV until (not) come.3sg.PF.AORIST sister your

"I was watching TV **until** your sister came"

In the second example negating subordinated VP has no impact on the sentence meaning. We propose an interpretation that covers all cases illustrated above with the exception of coordinating *dokato* properties. As a subordinated constituent, *dokato* clause belongs to the frame adverbials class. The interval referred to is construed depending on the aspectual structure provided by the verb. If possible, the end point of the interval is anchored. When a subordinated verb expresses a moment-like eventuality (i.e. points or achievements) or an eventuality composed by process and culmination/termination (accomplishments), it serves as an endpoint (sentences (1) and (2), positive verb form variant). If not, the interval is identified by the activity/state, that is, when during the interval a particular positive or negative situation holds (sentence (2), negative variant).

Eventualities and temporal ordering annotation was implemented within the CLaRK System. We used Constraints, XPath Insert and Transformation tools. Our first step was to add information about aspectual class sets. Then a number of constraints and regular grammars were applied in a particular order: identification of EVENT and SIGNAL elements, the relevant attributes that receive "sure" value, TLINK insertion, ELINK insertion, and finally, establishment of TRs type. This simple algorithm ends with ascribing VAGUE value where more and different kinds of data are needed to calculate relType.

We create algorithms for assigning one of the 4 possible relType values for TLINK in coordinated sentences: SIMULTANEOUS, IS_INCLUDED, ENDED_BY and VAGUE in cases where the rule-based approach is insufficient, and 3 possible values for ELINK: IS_INCLUDED, INCLUDES and SIMULTANEOUS. For subordinated sentences the number of possible values increases, and even expert annotators have difficulties accessing relType.

The results we obtained are the following. 166 TLINK and ELINK elements are inserted automatically. Overall we achieve 63.7 % recall, 91.1 % precision, F1-score – 0.75. The worst performance is for ordering bounded-bounded eventuality, where @AspCat = "acc-ach" or "ach" and one of the verbs is in a non-perfective tense, while the other is in perfect (disregarding the type of the sentence). Best performance was for ordering bounded-unbounded eventualities (in complex sentences).

5. Conclusions and further work

The annotation of eventualities and temporal relationships is a subtask of a more general project – annotation of temporal information (first time for Bulgarian language) on top of the BulTreeBank. The CLaRK System, the system originally used for the creation of the BulTreeBank, will be further employed for implementing TIMEX annotation. As a preliminary step, we have created a verb classification and a refined annotation tagset, based on the TimeML standard, which was tested by implementing algorithms for automatic temporal entities recognition and markup in the CLaRK system.

6. Acknowledgements

The author wish to express her gratitude to her supervisor Ruselina Nitsolova, as well as to Petya Osenova and Kiril Simov for their encouragements and guidance. Special thanks are due to the anonymous reviewers whose criticism helped to improve this paper substantially. The work was granted partially by LTfLL project (FP7-212578).

7. References

[1] Allen, J. Maintaining knowledge about temporal intervals. *Communications of the ACM*. 26:832-843 November 26, 1983.

[2] *Bulgarian Academy Grammar*. Abagar, Sofia, 1983.

[3] Damova, Mariana. *Tense and Aspect in Discourse: A study of the interaction between aspect, discourse relations and* temporal reference within discourse representation theory with special attention to Bulgarian. PhD thesis, Stuttgart, 1998.

[4] Freed, A. The Semantics of English Aspectual Complementation. D. Reidel P.Company, Dordrecht, Holland, 1979.

[5] Ivanova, K. Nachini na glagolnoto dejstwie v syvremennija bylgarski ezik. Izdatelstvo na BAN, Sofia, 1974.

[6] Laskova, L. Taksisni otnoshenija v bipredikativni izrechenija za vreme. MA Thesis. SU St. Kliment Ochridski, Sofia, 2003.

[7] Młynarczyk, A. Aspectual Pairing in Polish. Utrecht: LOT, 2004. Available at: http://igitur-archive.library.uu.nl/dissertations/2004-0309-140804/inhoud.htm Last accessed Jun 08, 2009

[8] Moens, M. & Steedman, M. Temporal Ontology and Temporal Reference. Computational Linguistics, 14(2):15-28, June, 1988.

[9] Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., & Pustejovsky, J. 2002. TimeML Annotation Guidelines, Version 1.2.1. Available at: http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf Last accessed: Jul 13, 2009.

[10] Simov, K., Osenova, P. & Slavcheva, M. 2004. BTB-TR03: BulTreeBank Morphosyntactic Tagset. BTB-TS version 2.0. Available at: http://www.bultreebank.org/TechRep/BTB-TR03.pdf Last accessed: Feb 36, 2009.

[11] Simov, K. & Osenova, P. BTB-TR05: BulTreeBank Stylebook. BulTreeBank Version 1.0. Available at: http://www.bultreebank.org/TechRep/BTB-TR05.pdf Last accessed: Feb 36, 2009.

[12] Simov, K., Peev Z., Kouylekov M., Simov A., Dimitrov M. & Kiryakov, A. CLaRK - an XML-based System for Corpora Development. In: Proc. of the Corpus Linguistics 2001 Conference, pp 558-560, 2003.

[13] Petruhina, E. Aspektual'nye kategorii glagola v russkom jazyke: v sopostavlenii s c'eshskim, slovatckim, pol'skim i bolgarskimi jazykami. Izdatel'stvo Moskovskogo universiteta, Moskva, 2000

Ambiguous Arabic Words Disambiguation: The results

Laroussi Merhben
UTIC(Monastir unit) higher school of techniques sciences of Tunis.
Aroussi_Merhben@hotmail.com

Anis Zouaghi
UTIC (Monastir Unit) superior Institute of informatics of Medenine
Anis.zouaghi@gmail.com

Mounir Zrigui
UTIC (Monastir unit) Faculty of sciences of Monastir
Mounir.Zrigui@fsm.rnu.tn

Abstract

In this paper we propose an hybrid system of Arabic words disambiguation. To achieve this goal we use the methods employed in the domain of information retrieval: Latent semantic analysis, Harman, Croft, Okapi, combined to the lesk algorithm. These methods are used to estimate the most relevant sense of the ambiguous word. This estimation is based on the calculation of the proximity between the current context (Context of the ambiguous word), and the different contexts of use of each meaning of the word. The Lesk algorithm is used to assign the correct sense of those proposed by the LSA, Harman, Croft and Okapi. The results found by the proposed system are satisfactory, we obtained a rate of disambiguation equal to 73%.

Keywords

Arabic ambiguous words, LSA, Harman, Okapi, Croft, Lesk algorithm, signatures and syntactic tagger.

1. Introduction

This work is part of the understanding of the Arabic speech [15]. In this paper we are interested in determining the meaning of Arabic ambiguous words that we can encounter in the messages transcribed by the module of speech recognition.

The word sense disambiguation (WSD) involves the association of a given word in a text or discourse with a definition or meaning (sense) which is distinguishable from other meanings potentially attributable to that word [12].

To assign the correct meaning, our method starts with the application of several pre-processing (tf × idf [14], normalization and syntactic tagging [2]) on words belonging to the context of the ambiguous word, subsequently we have applied the measures of similarities (Latent Semantic Analysis [5], Harman [8], Croft[3] and Okapi [13]) which will allow the system to choose the context of using the most closer to the current context of the ambiguous word, and we have applied Lesk algorithm [10] to distinguish the exact sense of the different senses given by this measures of similarity.

This paper is structured as follows, in section 2 we present the ambiguity of the Arabic language, after that in section 3 we describe the proposed method for disambiguation of ambiguous Arabic words later in section 4, we present the results of tests of our model.

2. Disambiguation of Arabic

The Arabic language is considered a difficult language to be automatically processed [10]. Among the characteristics that make this language processing ambiguous, we quote:

- The non vocalization of the Arabic language: a non vocalized Arabic word has several possible meanings. However, in modern editions, the texts in Arabic languages are not vocalized. We recall that vocalization in Arabic language is the addition of signs to the consonant to precise the pronunciation. Here is an example of a non-vocalized word: كتب (Kataba), this word might mean by way of his vocalization: كَتَبَ (he wrote), كُتُبُ (books), كُتِبَ (it was written). This phenomenal makes the problem of disambiguation more difficult;

- The structure of an Arabic word has a big problem for the automatic disambiguation. Indeed, an Arabic word can mean any expression in English or french. Here are some examples: the word وتتذكروننا (watatathakarounana) expresses the sentence in french “ and you remember us ", the same word (وبقوله) (wabikawlihi) which means in English" and by his word”. Thus the automatic understanding of such words requires a prior segmentation, a task that is not obvious;

- Another source of problems is the lack of language resources such as dictionaries, previously tagged corpus, and so on. This lack of resources with the characteristics of this language makes automatic processing more difficult;

In what follows, we describe the proposed method for disambiguation of the meaning of ambiguous Arabic words.

3. Proposed System

3.1 Method

Because of the lack of linguistic resources necessary for the automatic processing of the Arabic language, we preferred to use and test a non-supervised method. We note that Unsupervised methodology identifies patterns in a large sample of data, without the benefit of any manually labeled examples or external knowledge sources, on the other hand the supervised methodology Create a sample of training data where a given target word is manually annotated with a sense from a predetermined set of possibilities.

The Principe of our method is as follows: First, we started by collecting, from the web, various Arabic texts to

Student Research Workshop, RANLP 2009 - Borovets, Bulgaria, pages 45–52

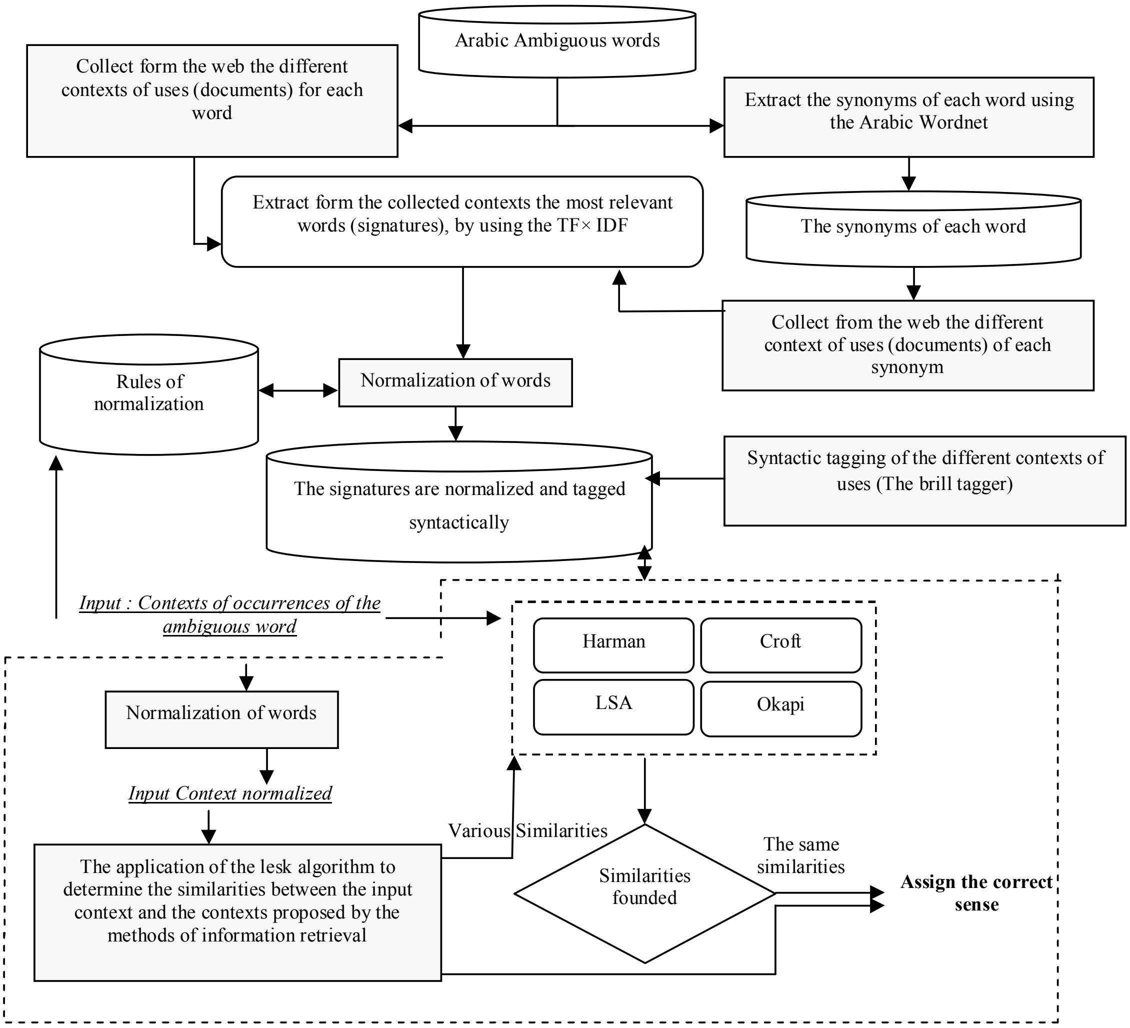

Arabic Ambiguous words
Collect form the web the different contexts of uses (documents) for each word
Extract the synonyms of each word using the Arabic Wordnet
Extract form the collected contexts the most relevant words (signatures), by using the TF× IDF
The synonyms of each word
Collect from the web the different context of uses (documents) of each synonym
Rules of normalization
Normalization of words
The signatures are normalized and tagged syntactically
Syntactic tagging of the different contexts of uses (The brill tagger)
Input : Contexts of occurrences of the ambiguous word
Harman
Croft
LSA
Okapi
Normalization of words
Input Context normalized
Various Similarities
The application of the lesk algorithm to determine the similarities between the input context and the contexts proposed by the methods of information retrieval
Similarities founded
The same similarities
Assign the correct sense

Figure 1. Method proposed to disambiguate the ambiguous Arabic word senses

build a corpus (see Section 4, Table 2) for several areas (i.e. sport, politics, religion, science, etc.).

From the corpus collected with the help of a linguist, we extracted the ambiguous words (words with several possible meanings out of context).

We note that we have applied several pre-processing steps (see Section 3.3) to the words that belonging to different contexts of use of the ambiguous word to improve the performance of the proposed system. We mean by context of use of an ambiguous word all sentences or texts in which the word has the same meaning.

From the Arabic WordNet [1] (lexical database of electronic Arabic words), we extract the synonyms of each word considered ambiguous. Then we collected the different contexts of use of these synonyms. This step enhances the number of contexts of use of each ambiguous word.

From the collection of possible contexts of use of each ambiguous word, and using the tf × idf measure [14] we were able to extract the different signatures, (the words that affect the meaning of each ambiguous word). Thus, each collection of signatures extracted from a context of use of an ambiguous word describes a unique sense of it. We also tested the contribution of syntactic knowledge on the outcome of disambiguation; we measured the similarity between the current context of an ambiguous word and its various contexts of use after tagging using the Brill tagger [2].

Following these pre-processing steps, we have implemented and tested several methods used in information retrieval: the latent semantic analysis [5], Harman [8], Croft [3] and Okapi [13], to measure the similarity between the current context of occurrence of the ambiguous word and the different possible contexts of use (possible meaning) of the word to disambiguate. The context which has a high similarity score with the current context is the most likely sense of the ambiguous word.

We note that we have tested these methods for measuring the similarity between the current context and all the possible contexts of use of each ambiguous word (Contexts represented in the form of texts and sentences) (see experimental results in section 4) in the first experiment we give the results obtained after pre-processing (Contexts represented by their signatures and tagged syntactically) Noticing that these methods do not always give the same result, we have tested the algorithm of Lesk to judge what is the most likely senses among those proposed by the methods listed above. In the following sub-paragraphs, we detail the different steps of the proposed disambiguation method. Figure 1 below describes our method.

3.2 Constitution of our Corpus

As mentioned previously, we have collected the various contexts of uses of each ambiguous word from the web and we do the same work for their synonyms that are obtained from a predefined lexical resource such as Arabic Wordnet [1]. With the help of linguistics we have given for each context the corresponding sense. Contexts (texts) are extracts of newspaper articles, which were recorded without restriction as to their nature and volume (see paragraph 1 in section 4).

All collected texts are non-vocalized. We used dictionary Al Wassit [6] to determine the definition of ambiguous words used for the test.

3.3 Pre-processing

3.3.1 Extraction of the signatures

The Several methods have been proposed to find for each given word the other words that appear generally next to him. In this experiment we have used the tf × idf measure (Term Frequency × Inverse Document Frequency) [14] it allows to assess the importance of a word in relation to a document, which varies depending on the frequency of the word in the corpus. For each context we take only the 20 words that have the maximum score (tf × idf), this encoding allows us to eliminate the stop words and the non-content words such as:

كان، له، فوق، حتّى، من، قد، بها، في،...

(he was, to him, on, to, from, then, with, whereas, ...)

These signatures represent the most basic part of our model because they represent the words that affect the meaning of each ambiguous word; these words have a higher likelihood of appearing together. If we don't find these signatures in the current context, in this case we extract from this context all the words that affect the meaning of ambiguous word and we add them to our database, this will ameliorate the performance of our system. Table 1 below shows some examples of signatures.

Table 1. Example of signatures describing the possible meanings of the word "عين" (ayn)

Different Senses	Number of signatures	Signatures
العين المبصرة (eye)	50	ترى ,الجسم ,الجمجمة ,تابعت ,النّور …, See, the body, the crane, follow, the light, ...
عين الماء الجارية (source of water)	46	الماء، الجبل، الضّيعة، الجارية، تسقي،... Water, mountains, the companion, which flows, baste,
الشيء عينه أي نفس الشيء (the same thing)	39	شيء, مسألة , حكم, منهج, الأصح ,… Thing, Problem, trial program, rather ,…
عينا على الأعداء أي جاسوسا (spy)	49	يتجسّس، يراقب، المفتّش، خائن،... Spy, monitor, inspector, traitor,
حرف العين (the word ayn)	56	حرف، اللّغة، كلمة، الجملة، تتركّب،... The letter, language, word, sentence, consists ...

3.3.2 Word Normalisation

In this step we group all the words that are derived from the same root in one cluster. The words that we have considered in this step are the signatures obtained previously. Our goal is to create a partitioning of set of data (signatures) into a set of relevant subclasses called clusters represented by a root. For example we take the words (ذهب (thahaba) - ذاهب (thaaheb) - مذهب (mathhab) - ذهبنا (thahabna)) (go, someone that will go, way, we went), all these words are derived from the same root ذهب. So, all the words are represented and replaced by the root ذهب. This grouping was done manually using linguists and dictionaries. We have thus constructed bags containing the words derived from each root. This treatment also contributed to the improved performance of the proposed system. Indeed disambiguation rate changed from 52% to 56% (see experimental results, Section 4).

3.3.2 Syntactic Tagging of contexts

To test the influence of syntactic knowledge on semantic disambiguation task, we tagged syntactically the different contexts of use of ambiguous words, using the transformation based learning in the Brill tagger [2]. Syntactic tags used in our experiment are three, they can indicate if the word in question is a particle, a verb or a noun. Syntactic tagging of the corpus allowed our system to study the contribution of syntactic information on the result of determining the correct orientation for each ambiguous term. To achieve this goal, we measured the similarity (see next paragraph) between the current context and contexts of use while taking into account the syntactic tags assigned to different words. The syntactic tagging system give a success rate of 78 %. This study has enabled us to obtain a gain of performance in terms of accuracy. Indeed disambiguation rate changed from 52% to 64.3%. (see experimental results, Section 4)

3.4 Estimation of the most relevant sense using LSA, Okapi, Harman and Croft

Let CC = m_1 m_2 … m m_{-1} … the context where the ambiguous word m appears. Suppose that S_1, S_2, .., S_k are the possible senses of m out of context. And CU_1, $CU_{2,}$… CU_K are the possible contexts of use of m for which the meanings of m are respectively: S_1, S_2, …S_K.

To determine the appropriate sense of m in the current context CC we have used the information retrieval methods (LSA, Okapi, Harman and Croft) which allow the system to calculate the proximity between the current context (Context of the ambiguous word), and the different use contexts of each possible sense of this word.

The result of each comparison is a score indicating the degree of semantic similarity (see equation 1) between the CC and CU given. This allows our system to infer the exact meaning of the ambiguous word. The following equation (1) describes the method used to calculate the score of similarity between two contexts:

$$S_t(CC, CU) = (\Sigma_{i \in RC} E(m_i) + \Sigma_{i \in LC} E(m_i)) / (\Sigma_{i \in RC} FE(m_i) + \Sigma_{i \in LC} FE(m_i)) \quad (1)$$

Where, $\Sigma_{i \in RC} E(m_i)$ et $\Sigma_{i \in LC} E(m_i)$ are respectively the sums of weights of all words belonging at the same time to the current context CC and to the context of use CU.

FE(mi), correspond to the first member of E(mi), where E (mi) can be replaced by one of the information retrieval methods : Croft, Harman or Okapi, whose equations are respectively:

- Harman measure [8]:

$$H(m) = W_H(m, CU(t)) = - \log (n(m) / N) \times [\log(n_{CU}(m) + 1) / \log(T(CU))] \qquad (2)$$

Where, WH(m, CU(t)) is the weight attributed to m in the use contexts CU of the ambiguous word t by the Harman measure ; n(m) is the number of the use contexts of t containing the word m ; N is the total number of the use contexts of t ; $n_{CU}(m)$ is the occurrence number of m in the use context CU ; and T(CU) is the total number of words belonging to CU.

- Croft measure C(m)[3]:

$$C (m) = W_C(m, CU (t)) = - \log (n(m) / N) \times [k + (1\text{-}k) \times (n_{CU}(m) / Max_{x \in CU}\, n_{CU}(x))] \quad (3)$$

Where, W_C(m, CU (t)) is the weight attributed to m in the user context CU of t by the Croft measure; k is a constant that determines the importance of the second member of C(m) (here, k = 0,5) ; and and $Max_{x \in c}$ $n_{CU}(x)$ is the maximal number of occurrences of word m in CU.

- Okapi Measure [13]:

$$O(m) = W_O(m, CU(t)) =$$

$$\log [(N - n(m) + 0,5) / n(m) + 0.5] \times [n_c(m) / (n_{CU}(m) + (T(CU) / T_m(B)))] \quad (4)$$

Where, W_O(m, CU(t)) is the weight attributed to m in CU of t by the Okapi measure ; and Tm(B) is the average of the collected use contexts lengths.

- Latent Semantic Analysis [5]:

After the construction of the matrix A (term × documents) LSA find an approximation of the lowest rank of this matrix, by using the singular value decomposition which reduce obtains N singular values, where N = min (number of terms, number of docs). After that, the K highest singular values are selected and produces an approximation of k-dimension to the original matrix (It's the semantic space) In our experiments we used the Cosine to compare the similarities in the semantic space and k = 8.

3.5 Applying the Lesk algorithm to assign the correct sense

We adapted the Lesk algorithm [11] to calculate the proximity between the words that appear in the different definitions given by the methods used previously and the current context. The input of the algorithm is the word t and S = (s_1, ..., s_N), are the candidates senses corresponding to the different contexts of use achieved by applying methods of information retrieval. The output is the index of s in the sense candidates.

The lesk algorithm simplified [7] :

```
Begin
   Score ← 0
   Sense ← 1 // Choose the sense
   C ← Context (t) // Context of the word t
   For all I ∈ [1, N]
       D ← description (si)
   Sup ← 0
   For all w ∈ C do
       w ← description (w)
       sup ← sup + score (D, w)
   if sup > score then
     Score ← sup
      Sense ← i
End.
```

The choice of the description and context varies for each word tested by this algorithm.

The function Context (t) is obtained by the application of the input context. The function description (s_i) finds all the candidate senses obtained by the information retrieval methods. The function score return the index of the candidate sense to take: score (D, w) = Score (description (s), w).

The application of this algorithm allowed us to obtain a rate of disambiguation up to 73% (see paragraph 3 in section 4).

4. Experimental results

4.1 Characteristics of our corpus

The table 2 below describes the size of the corpus collected representing all contexts of use (texts) of ambiguous words considered in our experiments. We note that we intend to increase the size of the corpus in our next experiments.

Table 2. Characteristics of the collected Corpus

Total size of the corpus	1900 texts
Number of ambiguous words	10 words
Average number of synonyms of each ambiguous word	4
Average number of the possible senses	5
Total number of contexts of uses	300 texts
Average size of each context of use	560 words, 40 sentences

All the methods that were applied by our system consider all this characteristics of corpus in the different tests. We note that the corpus is manually created and evaluated.In our experiments we have used 10 ambiguous words to test our model. The Table 3 below describes an example of some contexts of use of the ambiguous word "الظّلمات" (atholoumat) for each sense.

Table 3. Example of context of use for each sense of the ambiguous word "الظلمات" (atholoumat)

Sense	Example of a contexts of use used in the test
انعدام الضّوء darkness	....سميت تفاعلات **الظلام** بعملية التمثيل الضوئي بسقوط الضوء على مجموعة من الخلايا ... The reactions of darkness called photosynthesis of the light because of the concentration of light on a set of cells
الجهل ignorance	... مشاعل هذه الحضارة الفتية تبدد **ظلمات** الجهل من خلال التمدن الإسلامي... The occupation of youth culture dissipates the darkness of ignorance to the Islamic civilization
العمى Blind	... يمكن ان يؤدي إلى ضعف البصر الدائم أو إلى **ظلمات** العمى May cause permanent visual impairment or a darkness of blindness

In table 4 below, we will give an example of data tested by our model and the sense given by every method. The data tests are randomly created.

Table 4. Results given by disambiguating the word "عين" in an example of test data

Example of test data	Sense affiliated				
	LSA	**Harman**	**Croft**	**Okapi**	**Lesk**
تبدو عين الإنسان كرويّة الشّكل	العين المبصرة	العين المبصرة	العين المبصرة	حرف العين	العين المبصرة
The eye of the human appear like a spherical form	eye	eye	eye	The word ayn	eye

4.2 Comparison of results obtained by the methods of information retrieval:

The figure 2 below presents the results obtained by using the methods ASL, Okapi, Croft and Harman. We note that we used the following metric to measure the rate of disambiguation:

Exact rate = (Number of senses obtained correctly / Number of senses assigned) × 100

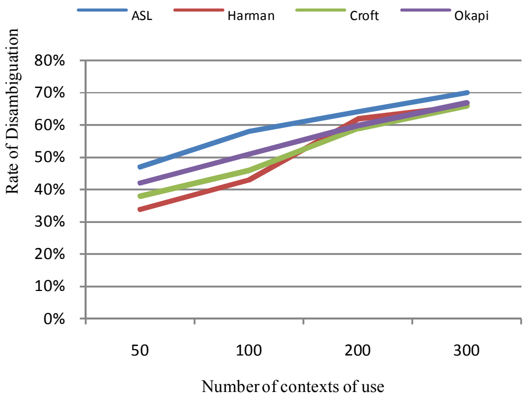


Figure 2. Comparison of results obtained by the disambiguation methods: LSA, Harman, Okapi and Croft

The results presented in Figure 2 above show that a bad disambiguation is obtained whenever there is a lack of contexts of use. Indeed, we note that the rates of disambiguation become satisfactory when the number of contexts of use is equal to 300. The worst results are obtained when the number is less than 100.

We can therefore conclude that the lowest rate of disambiguation is mainly due to the insufficient number of contexts of use, which result in the failure to meet all possible events. We also note that LSA provides the best results.

4.3 Experiment 1: Results obtained by the application of the LSA, Okapi, Harman, Croft and Lesk algorithm

The Table 5 below shows the rates of disambiguation obtained corresponding to ten Arabic ambiguous words. We validate results with 25 randomly selected samples. We note that the proposed hybrid system successfully disambiguate 76% of ambiguous words.

Table 5. Rate of disambiguation of arabic ambiguous words after pre-processing (extraction of signatures, word normalization and syntactic tagging)

Ambiguous words	Rate of sense affiliated correctly (%)				
	LSA	Harman	Croft	Okapi	Lesk
عين (ayn)	74	65	67	62	68
حسب (hasaba)	64	57	59	58	69
هان (hana)	62	54	49	52	74
الظلمات (atholoumat)	83	75	73	71	81
النّور (annour)	78	72	70	71	78
شعر (cheer)	69	65	63	61	70
فجر (Fajara)	57	51	51	53	62
نبع (nabaa)	71	62	60	60	73
دجم (dajama)	68	65	66	66,5	72
عقل (aakl)	75	65	61	64	84
the rate of disambiguation (%)	70.1	63.1	61.9	55.2	73.1

From Table 5, we note that the rate of disambiguation of the word "عين" (ayn) is lower than the other words, since it has more senses and more signatures, which makes the disambiguation of the term complex than the other words. Figure 3 below shows the influence of the number of signatures on the rate of disambiguation obtained.

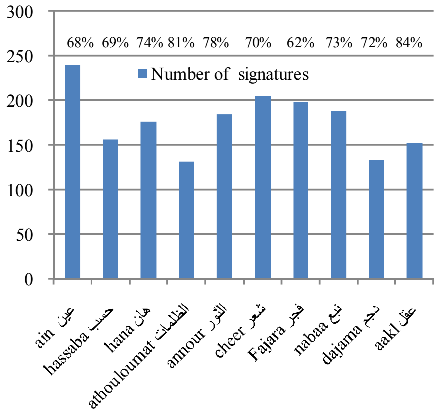


Figure 3. Influence of the number of signatures of the ambiguous words on the rates of disambiguation

We also note that the results obtained by the methods used in information retrieval: Harman, Croft and Okapi are very close.

The average rate of disambiguation is equal to 60%.

While disambiguation results obtained from the latent semantic analysis are often different from those found by Harman, Croft and Okapi. The average of disambiguation obtained by LSA is equal to 70.1%. We can then infer that the LSA gives better results. After some tests it was noted that these measures do not have in all cases the same meaning to be assigned (see Table 6 below). This makes the system unable to make a decision on the correct orientation. This explains why we decided to use the algorithm of Lesk. This algorithm allows our system to improve the results (we have achieved an average of disambiguation equal to 73%), it allow the system to choose the adequate sense.

Table 6. Example of the results of test of the word "هان" (hana) in the sentence:

"أكبر **مهانة** يتعرض لها الفرد عندما يعبد حجرا أو شجرا أو حيوانا أو يخضع لبشر
حي أو ميت"

(The greatest **humiliation** that a person may encounter when making a prayer to a stone or a tree or an animal or it becomes the subject of a human being living or dead.)

Ambiguous word	LSA	Harman	Croft	Okapi	Lesk
هان	ذلّ	سهل	رخص	سهل	ذلّ
hana	humiliate	simplify	Lowering	simplify	humiliate

4.4 Experiment 2: Results obtained before pre-processing

In this experiment we have tested the influence of the use of signatures on the results of disambiguation of the meaning of a word. Table 7 below shows that the rates of disambiguation of Arabic words obtained using the contexts of use without going through the signatures are increased from 52% to 73%.

Table 7. The rate of disambiguation of ambiguous words before pre-treatement

Ambiguous words	**Rate of sense affiliated correctly (%)**				
	LSA	**Har man**	**Croft**	**Okapi**	**Lesk**
عين(ayn)	42	36	37	36	48
حسب (hasaba)	50	43	45	41	54
هان (hana)	42	39	37	36	45
الظلمات (atholoumat)	62	51	54	53	64
النّور (annour)	51	42	41	41	54
شعر (cheer)	46	34	34	33	49
فجر (Fajara)	41	38	40	40	46
نبع (nabaa)	53	45	46	47	49
دجم (dajama)	45	36	34	34	48
عقل (aakl)	59	56	57	56	62
the rate of disambiguation (%)	49.1	42	42.5	41.7	52

4.5 Experiment 3: Studying the influence of the syntactic knowledge

We also tested the contribution of syntactic knowledge on the obtained results. For that, we have used the Brill tagger (see section 3.4). The table 8 shows that the rates of disambiguation of Arabic words obtained before syntactic tagging of contexts of use of each ambiguous word decreased compared to Experiment 1 (using syntactic tags), the rate decreased from 73% to 64.3%, while it was increased compared to the previous experiment 2 (use of contexts as they are, without the use of signatures) from 52% to 64.3%.

Table 8. The rate of disambiguation using syntactic tags

Ambiguous words	**Rate of sense affiliated correctly (%)**				
	LSA	**Har man**	**Croft**	**Okapi**	**Lesk**
عين(ayn)	42	36	37	36	48
حسب (hasaba)	63	56	58	60	59
هان (hana)	57	53	51	50	60
الظلمات (atholoumat)	50	49	46	47	57
النّور (annour)	72	64	61	64	74
شعر (cheer)	67	54	59	55	71
فجر (Fajara)	61	56	54	54	67
نبع (nabaa)	51	48	47	49	58
دجم (dajama)	65	62	62	60	68
عقل (aakl)	56	49	48	46	61
the rate of disambiguation (%)	65	62	63	63	70

4.6 Comparison of the proposed hybrid system with other systems disambiguation:

In this part we do a comparison of the results founded by our system with other system of disambiguation, comparing these results with the various works is a difficult task, because we do not work on the same corpus, or the same language, or with the same methods:

The method created by Lesk [11] used a list of words appearing in the definition of each sense of the ambiguous word achieved 50% - 70% correct disambiguation, Our system achieved 73% correct disambiguation

Karov and Edelman [9] (in this issue) propose an extension to similarity-based methods which gives 92% accurate results on four test words.

5. Conclusion

We have proposed a system for disambiguation of words in Arabic. This system is based simultaneously on the methods of information retrieval and the algorithm of Lesk used to calculate the proximity between the current context (i.e. the occurrence of ambiguous word) and the different contexts of use of the possible meanings of the word. While Lesk algorithm is used to help the system to choose the most appropriate sense proposed by previous methods. The results founded are satisfactory. For a small sample of 10 ambiguous words, the proposed system allows to determine correctly 73% of ambiguous words. We have tried to establish a sufficiently robust system based on methods that have improved their success in many system of word disambiguation. On the other hand, during the pre-processing we tried to make the ambiguous Arabic words known by the system we proposed a database containing the possible contexts of use for each sense of an ambiguous word, synonyms, signatures identifying the meaning of each one and syntactic tags.

We propose that in the future works we can use a multi-agent system that takes the more appropriate result given by all the methods applied by our system.

6. References

[1] Black, W. J. & Elkateb, S. (2004) A Prototype English-Arabic Dictionary Based on WordNet, Proceedings of 2nd Global WordNet Conference, GWC2004, Czech Republic: 67-74.

[2] BRILL E. (1993), A Corpus-Based Approach to Language Learning, Thesis non-published, University of Pennsylvania, Department of Computer and Information Science.

[3] Croft.W, 1983. Experiments with representation in a document retrieval system; Research and development, 2(1) ; pp. 1-21 ; 1983.

[4] De Loupy, 2000. Assessing the contribution of linguistic knowledge in semantic disambiguation and information retrieval. THESIS presented in the University of Avignon and the country of Vaucluse.

[5] Derwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshmann, R. 1990. Indexing by Latent Semantic Analysis. Journal of the American Society for Informartion Science, 41 : 391-407.

[6] Dictionary Al Wasit, 4th Edition, 2003, Academy of the Arabic language, Sunrise International Library.

[7] Florentina Vasilescu , 2003. Monolingual corpus disambiguation by the approaches of Lesk : University of Montreal, Faculty of Arts and Sciences; Paper presented at the Faculty of Graduate Studies to obtain the rank of Master of Science (MSc) in computer science.

[8] Harman D., 1986. An experimental study of factors important in document ranking ; Actes de ACM Conference on Research and Development in Information Retrieval ; Pise, Italy ; 1986.

[9] Karov, Yael and Edelman, Shimon (1998). “Similarity-based word sense disambiguation”. In this issue.

[10] Larkey L. S., Ballesteros L. and Connell M., « Improving Stemming for Arabic Information Retrieval : Light Stemming and Cooccurrence Analysis », In Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), Tampere, Finland, August 2002, p. 275-282

[11] Michael Lesk, Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone , ACM Special Interest Group for Design of Communication Proceedings of the 5th annual international conference on Systems documentation, p. 24 - 26, 1986. ISBN 0897912241

[12] Nancy Ide, Jean Verronis. 1998.Word Sense Disambiguation: The State Of The Art. Computational Linguistics, 2424:1, 1-40.

[13] Robertson et al., 1994.: S. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford ; Okapi at TREC-3 ; Acts de Third Text Retrieval Conference (TREC-3), NIST special publication 500-225 ; pp. 109-126 ; Gaithersburg, Maryland, USA ; 1994.

[14] Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), 513-523.

[15] Zouaghi A., Zrigui M., Antoniadis G. 2008. Understanding of the Arabic spontaneous speech: A numeric modelisation, Revue TAL VARIA.

ZAC.PB: An Annotated Corpus for Zero Anaphora Resolution in Portuguese

Simone Pereira
University of the Algarve
Campus de Gambelas
P-8005-139 Faro, Portugal
simonecp@gmail.com

Abstract

This paper describes the methodology adopted in the construction of an annotated corpus for the study of zero anaphora in Portuguese, the ZAC corpus. To our knowledge, no such corpus exists at this time for the Portuguese language. The purpose of this linguistic resource is to promote the use of automatic discovery of linguistic parameters for anaphora resolution systems. Because of the complexity of the linguistic phenomena involved, a detailed description of the different situations is provided. This paper will only focus on the annotation of subject zero anaphors. The main issues regarding zero anaphora in Portuguese are: indefinite subjects, either without verbal agreement marks or with first person plural or third person plural verbal agreement; position of the anaphor relative to its antecedent, i.e. anaphoric and cataphoric relations; coreference chains inside the same sentence and spanning several sentences; and determining the head of the antecedent noun phrase for a given anaphor. Finally, preliminary observations taken from the ZAC corpus are presented.

Keywords

Anaphora resolution, zero anaphora, corpus linguistics, corpus annotation, syntax, Brazilian Portuguese.

1. Introduction

In many linguistic situations, redundant NPs, usually already present in a previous utterance or in a previous constituent of the same utterance may be reduced to pronoun or to zero (NP deletion) in order to avoid redundancy [1].

(1.1) **John went to school and then John went to the mall*

(1.2) *John went to school and then* [*he went*] *to the mall*

Portuguese has a very rich verbal inflection, and the subject can easily be recovered through verbal inflection.

The grammatical rules governing NP deletion may vary among languages, even among different varieties of the 'same' language, as in the case of Brazilian (^{bp}) vs. European Portuguese (^{ep}). For example, the Portuguese equivalent for the examples (1.1)-(1.2) should be:

(1.3) **O João*$_i$ *foi à escola e depois o João*$_i$ *foi ao* ^{ep}*centro comercial/*^{ep,bp}*shopping*

(1.4) *O João*$_i$ *foi à escola e depois* (ε + ^{*ep,bp}*ele*$_i$) foi ao ^{ep}*centro comercial/*^{ep,bp}*shopping*

(1.5) *O João*$_i$ *foi à escola e depois* ao ^{ep}*centro comercial/*^{ep,bp}*shopping*

In the previous examples, the reduction of the verb imposes the subject NP deletion; otherwise it can be reduced, in Brazilian Portuguese, both to pronoun and to zero, while in European Portuguese only zero-reduction is allowed.

In order to correctly resolve zero anaphora [2], NLP systems require (a) the correct identification of the zero anaphor and (b) the correct identification of the antecedent of the zero anaphor[1]. Several strategies can be used to achieve this goal. For machine learning techniques, an annotated corpus is required.

This paper describes the methodology adopted in the construction of an annotated corpus for the study of zero anaphora in Portuguese, the ZAC corpus. To our knowledge, no such corpus exists at this time for the Portuguese language. The purpose of this linguistic resource is to promote the use of automatic discovery of linguistic parameters for anaphora resolution systems[2]. Our ultimate goal is to implement a module for zero anaphora resolution in the Portuguese grammar [3] developed under **X**erox **I**ncremental **P**arser (XIP) [4].

Because of the complexity of the linguistic phenomena involved, a detailed description of the different situations is provided. This paper will only focus on the annotation of subject zero anaphors. The main issues regarding zero anaphora in Portuguese are: indefinite subjects, either without verbal agreement marks or with first person plural or third person plural verbal agreement; position of the

[1] For clarity, *anaphor* is used to designate the pronoun, in NP reduction, or the syntactic slot left empty by NP deletion, while *anaphora* is a general term for the referential relation between the anaphor and its antecedent. It includes both *anaphora* proper, if the antecedent appears in a previous moment in discourse and *cataphora* if it appears after later moment.

[2] A similar corpus has been presented for Spanish [5] but in a different theoretical framework. A corpus for anaphora resolution has been produced for Brazilian Portuguese [6], but as far as we know only coreference chains between anaphors have been annotated, and no information was available for zero anaphors. Adaptation of the Mitkov algorithm [2] to Brazilian Portuguese pronoun resolution is given in [9].

Student Research Workshop, RANLP 2009 - Borovets, Bulgaria, pages 53–59

anaphor relative to its antecedent, i.e. anaphoric and cataphoric relations; coreference chains inside the same sentence and spanning several sentences; and determining the head of the antecedent noun phrase for a given anaphor. Finally, preliminary observations taken from the ZAC corpus are presented.

2. Building the corpus

To our knowledge, there is no available corpus marked up with deleted subject NPs. Because of this lack on linguistic resources, an annotated corpus has been built for this study. The corpus consists on a set of full and partial texts retrieved from the web, and digitalized from books, encompassing several genres, namely journalistic and literary text from contemporary authors. The corpus is provided in text format, but the annotation adopted can be easily converted into other formats. Table 1 shows the breakdown per genre type of the ZAC corpus current content.

Table 1. Content of the ZAC corpus

	ZAC corpus	
Text types	words	%
Special Report	15.791	45%
News	1.769	5%
Chronicle	8.385	24%
Fiction (short story)	3.227	9%
Fiction (romance)	6.040	17%
Total	35.212	

3. Annotating the corpus

The corpus was annotated by two annotators working together. General notation is as follows: Zero anaphors are marked by a zero symbol '0' inside brackets [], followed by an equal sign '=' and the arrow symbols '<' and '>', corresponding to anaphora and cataphora relations, respectively, and a word indicating the head of the antecedent noun phrase (NP).

3.1 Annotated cases

a) deleted subject

Only deleted subject of non-auxiliary verbs are to be marked. Verbal chains with auxiliary verbs whose subject has been zeroed count as a single verb form, hence there will be only one anaphor marked (3.1):

(3.1) *Mais de 90% dos machos descendentes das cobaias apresentavam os mesmos problemas, sem nunca* [0=<machos] *terem sido expostos ao inseticida*

Over 90% of male descendants of the [experiment] subjects showed the same problems without ever [males] having been exposed to insecticide

In coordinated clauses only the zeroed subject of explicit verb forms is marked (3.2):

(3.2) *O profeta o obsedia e* [0=<profeta] *o persegue tanto que* [0=<profeta] *o vê em todo lugar;* [0=<<profeta] *preenche literalmente a paisagem, o que torna a ilusão visual…*

The prophet obsesses him and [he=the prophet] pursues him so much that he sees him everywhere; [the prophet] literally fills the landscape, which makes the visual illusion…

If the zeroed subject refers to a subordinate clause, then the anaphor will be noted [0(clause)=X] where X indicates the main verb of the antecedent clause (this situation is relatively rare in the corpus) (3.3):

(3.3) *"Esconder um programa desta magnitude não é apenas inapropriado, mas* [0(clause)=esconder] *é também ilegal", disse o senador democrata Dick Durbin.*

"Hiding a program of this magnitude is not only inappropriate but [it] is also illegal," said democratic senator Dick Durbin.

However, in some sentences, the reduced material cannot be easily recovered from the preceding discourse, hence, even if the anaphor type may be indicated, the antecedent proper is left unknown '?'.

On coordinated relative clauses, where the second relative pronoun has been zeroed, it is marked with the special notation [0(que)=<X], where X represents the antecedent of the relative pronoun (this situation is not frequent in the corpus) (3.4):

(3.4) *Os processos epigenéticos também podem ocorrer pela modificação das histonas, as linhas que envolvem o DNA e* [0(que)=<linhas] *formam um novelo*

The epigenetic processes can also occur by the modification of histones, the lines that involve the DNA and form a ball

b) noun phrases

For NPs whose head is a nominal determiner, for example *conjunto* 'set' (3.5) and *maioria* 'majority' (3.6), it is this head noun that the zeroed anaphor is referred to, even if the semantic head of the noun phrase is the complement of that determiner

(3.5) *O terceiro fenômeno epigenético consiste na ação dos micro-RNAs, um conjunto de nucleotídeos que percorre o genoma* [0=<conjunto] *ligando e* [0=<conjunto] *desligando os genes*

The third epigenetic phenomenon consists in the action of micro-RNAs, a set of nucleotides that travel the genome connecting and disconnecting the genes

(3.6) *Já as garotas tiveram resultados melhores: 75% dos homens toparam no ato. Dos 25% restantes, a maioria pediu desculpas,* [0=<maioria] *explicando que* [0=<maioria] *tinha marcado de* [0=<maioria] *sair com a namorada*

On the other hand the girls had better results: 75% of men immediately agreed. From the remaining 25%, the majority apologized, explaining that [they] already had a date with their girlfriend

In the case of compound nouns, only the head noun is to be referred to in the zeroed anaphor. Because of tokenization criteria we use, prefixed nouns are considered compound words (e.g. *ex-colegas* 'ex-partners') (3.7):

(3.7) *Um exemplo conhecido dos adeptos do Orkut no Brasil são os ex-colegas de escola que, depois de anos sem* [0=<ex-colegas] *se comunicar e mesmo sem* [0=<ex-colegas] *ter nenhuma afinidade pessoal, [0=<ex-colegas] passam a engordar a lista de amigos virtuais uns dos outros*

A known example of Orkut supporters in Brazil are the ex-school mates who, after years without communicating, even without having any personal affinity, start engrossing the list of each other's virtual friends

Compound pronoun *a gente*, corresponding to a first person plural 'we', but imposing a third singular verbal agreement, is to referred to by the form *gente* (3.8):

(3.8) *— Mas a gente queria* [0=<gente] *ver filme, não show*

— But we wanted to see a film, not a show

The same happens with indefinite pronoun *todo (o) mundo* 'everyone', which will be referred to by the head noun *mundo* (3.9):

(3.9) *E nem todo mundo aprendeu a* [0=<mundo] *usá-los a seu próprio favor*

And not everyone learned how to use them to their own advantage

Other compound (frozen) expressions (3.10), syntactically non-analyzable, and half-frozen expression with infinitives (3.11) are left without notation:

(3.10) *[…] genes […]. São eles que ensinam aos outros genes* o caminho a seguir, *para* [0=<eles] *dar continuidade às espécies [...]*

[…] genes […]. It is them that teach others genes the way forward, in order to give continuity to the species

(3.11) No decorrer das décadas*, no entanto, a população acabou se aprofundando na miséria.*

Over the decades, however, people just went deeper into poverty

Compound proper names (named entities, in majuscules) are considered a single token and therefore, will be referred to in the notation of zero anaphors. In the case of titles in apposition with proper names, the two elements are considered together as the head of that NP (e.g. Dona Marta 'Mrs Marta').

In the case of coordinated antecedent NPs or PPs, only the first head noun is to be referred to by the zero anaphor, but with the special notation ',&' after that head noun.

With the so-called pronominal use of definite and indefinite articles, as well as with demonstrative pronouns, the zeroed noun is not to be referred to in the following zero anaphor and hence a pronominal analysis is adopted for these words (3.12):

(3.12) *E os demais, apesar de* [0=<os] *serem titulados, terão de ter experiência profissional na área do curso.*

And the remaining [students], although [they] have already graduate, will have to acquire professional experience in the course's area

c) indefinite subject

The indefinite subject is annotated as [0=indef] (3.13):

(3.13) [0=indef] *Nascer com patrimônio genético idêntico não significa que as pessoas crescerão tendo corpo, mente e doenças iguais*

To be born with identical genetic heritage does not mean that people will grow up with similar body, mind and disease

Indefinite elliptical subject where there is a systematic ambiguity with first person plural *nós* 'we', will be specially noted [0=1p] (3.14):

(3.14) *As descobertas são impressionantes.* [0=1p] *Conseguimos informações preciosas sobre os genes, as marcas epigenéticas e as mudanças do genoma ao longo da vida, o que dá início a uma revolução*

The findings are impressive. We got valuable information about the genes, the epigenetic markings and the changes of the genome throughout life, which initiates a revolution

In this example, the first person plural may correspond to: a) a real plural, referring to the speaker and his/her team of researchers; b) a modesty plural, referring to the speaker; or c) the indefinite (generic) subject, referring to the scientific community as a whole. Naturally, such ambiguities cannot be resolved at this stage.

Sentences with zeroed subject and with the verb in the third-person plural will be annotated [0=3p]; this type of subject is systematically ambiguous between: a) indefinite subject with the particular (empowered) connotation; and b) a simple third-person plural, only context can disambiguate it (3.15):

(3.15) *"Ainda* [0=3p] *estão fazendo isso lá embaixo",* [0=<<Zé Lopes] *acrescenta, sobre as praias sem vigilância ao longo do Rio Jutaí, um afluente do Solimões*

"[They] are still doing it down there," [Zé Lopes] adds, speaking about the beaches without surveillance along the Jutaí river, a tributary of the Solimões

In case the antecedent of a zero anaphor cannot be precisely determined, a question mark will be used instead `[0=?]` (3.16):

(3.16) *O encontro acontecera de repente, mas* `[0=?]` *era como se* `[0=3p]` *já tivessem sido amigos a vida inteira.*

The meeting happened suddenly, but [it] was as if [they] has been friends for [their] entire life

d) impersonal subject

The impersonal subject is annotated as `[0=impers]`. This notation may cover different syntactic and semantic structures, such as: impersonal constructions[3] with verbs *ter* (in BP) and *haver* (both in BP and EP); meteorological constructions; temporal expressions [7].

3.2 Coreference chains

A coreference chain is established between a sequence of anaphors and their antecedent noun phrase. When the antecedent of a zero anaphor is in a previous sentence[4], the notation `[0=<<X]` is used. The zero anaphor will be marked `[0=<<X]`, no matter how many sentences away it may be (3.17):

(3.17) *Os participantes concordaram com um programa ousado de combate à deterioração da terra, do ar e da água. Também* `[0=<<participantes]` *decidiram* `[0=<<participantes]` *buscar o crescimento econômico sem* `[0=<<participantes]` *degradar o meio ambiente*

The participants agreed on a bold program for combating the deterioration of land, air and water. [They] also decided to pursue economic growth without degrading the environment

However, if in the discourse the first-person plural is used as an indefinite and there is no necessary coreference chain between two (far apart) `[0=1p]` instances, the signs for anaphoric (<)/cataphoric (>) relation are not used.

In a coreference chain within the same sentence, if the antecedent of a zero anaphor 0_2 is also another zero anaphor 0_1, the head of the antecedent NP of the later 0_1 is repeated.

In certain cases, a coreference chain can be determined among indefinite subjects; in this (relatively rare) situation, the coreference relation is marked `[0=<indef]`. The same happens with other indefinite subjects, such as the first-person plural (1p), and the third-person plural (3p).

3.3 Excluded cases

In the annotation, some cases were excluded.

a) adjectives

The subject of adjectives is only marked if they appear with their copula verb (e.g. *ser*, *estar*, 'to be') (3.18). Therefore the zeroed subjects of adjectives in apposition are not marked (3.19):

(3.18) *O mundo científico ficou ainda mais complexo depois do mapeamento genético feito há seis anos, quando os pesquisadores passaram a se dedicar a entender a função de cada um dos genes e, o supremo desafio,* `[0=<pesquisadores]` *explicar as razões pelas quais eles às vezes exercem suas funções e outras* `[0=<eles]` *parecem hibernar preguiçosamente nos cromossomos sem nunca* `[0=<eles]` ser <sic> ativados *[…]*

The scientific world became even more complex after the genetic mapping made six years ago, when the researchers began to devote themselves to the understanding of the function of each gene and, the ultimate challenge, to explain the reasons why they sometimes perform their functions and other times they seem to hibernate lazily in the chromosomes without ever being activated

(3.19) *Ela ajudará na criação de remédios personalizados,* capazes *de* `[0=<remédios]` *alterar o genoma para* `[0=<remédios]` *deter o desenvolvimento de doenças e de transtornos psíquicos*

It will help in the creation of personalized medicine, capable of altering the genome in order to halt the development of diseases and mental disorders

b) past participle

The past participle is considered as an ordinary adjective and its zeroed subject should be marked accordingly depending on the presence (3.20) or absence (3.21) of the copula verb.

(3.20) *Certamente* `[0=<marido]` estava *armado*

Certainly the husband was armed

(3.21) *Hoje, líderes indígenas* formados *em universidades dirigem entidades e* `[0=<líderes]` *se espelham em Evo Morales, o índio aimará que preside a Bolívia. (no mark-up)*

Today, indigenous leaders trained in universities lead several institutions and [feel that they] are mirrored in Evo Morales, the Aymara Indian who presides over Bolivia

The past participle is considered a verbal form when it makes part of a compound tense with auxiliary verbs *ter* 'to have' (3.22) or (rarely) *haver* 'to there be' (3.23):

[3] Impersonal constructions may also appear with a NP and a gerund (BP/EP) or a prepositional infinitive (only in EP): [0=impers] ^{bp,*ep}Tem/^{bp,ep}Há gente ^{bp,ep}fazendo/^{*bp,ep}a fazer isso 'There is people doing this'.

[4] The separators ';' and ':' are considered sentence boundaries, along with other common sentence separators ('.', '?', '!', etc.).

(3.22) *"Eles precisam de tempo e de intimidade; como diz o ditado,* `[0=<eles]` *não podem se conhecer sem que* `[0=<eles]` tenham comido *juntos a quantidade necessária de sal"*

“They need time and intimacy; as the saying goes, [they] cannot cannot know each other without having eaten together the necessary quantity of salt”

(3.23) *Apesar de* `[0=>Arthur]` haver errado *todos os seis tiros, Artur conseguiu afastar a criatura.* `[0=<Arthur]` *Ajudou o senhor José a levantar*

Although Arthur had failed all six shots round, he managed to keep the creature away. [He] helped Mr. José to stand up

c) reduced gerundives

Like adnominal and appositional adjectives, in reduced gerundives resulting from relative clauses the subject is considered to be explicit and it is not marked (3.24). Otherwise gerundive adverbial clauses need the marking of zeroed subjects (cfr. (3.5)):

(3.24) *Luiz percebeu faíscas* saindo *de um poste à frente da casa*

Luiz saw sparks coming out of a pole in front of the house

d) topicalization structures and other forms of focus

Topicalization structures and other forms of focusing sentence elements involving changes in sentences’ basic word-order are not marked and the syntactic position left empty by the moved constituent is not signaled (3.25):

(3.25) *De fato pesava bastante, o tal saco*

Indeed [it] weighed a lot, that bag

In much the same way, cleft sentences with *ser … que* are not marked for their subject NPs (3.26):

(3.26) *É nas trilhas desse vazio,* `[0=>aventureiros]` *desfraldando falsas bandeiras do progresso, que aventureiros nacionais e internacionais invadiram a floresta e* `[0=<aventureiros]` *desataram as tragédias*

It is in the trails of this gap, unfurling the false flags of progress, that the national and international adventurers have invaded the forest and have untied the tragedies

e) direct speech, imperative, interrogative and exclamative sentences

In the case of direct speech (for example, in interviews) the first-person subject and the second-person (eventually the *você* personal pronoun, corresponding to a second-person but imposing to the verb a third-person agreement), if zeroed, are not to be marked.

In much the same way, the zeroed subject of imperative sentences; direct, total (yes/no) or partial (*wh-*); interrogative sentences; question tags; and exclamative sentences, where the speaker or the addressee are integrated in the discourse, are not to be marked. For indirect interrogative subordinate clauses with interrogative *qu-* (*wh-*) pronouns (*question cachée*), the pronoun is considered the head of the clause and can be referred to by a zero anaphor (3.27):

(3.27) *— É essa casa aqui.* `[0=?]` *Estão ouvindo?*

— This here is the house. Are [you_3PL] listening?

f) causative operator verbs

On constructions of causative operator verbs [8] with restructured subject, the structurally zeroed slot of the subject of the dependent clause is not marked (3.28):

(3.28) *A falta de comunicação com o resto da Terra* permitiu ao regime permanecer *mergulhado no passado (subject of permanecer is not marked)*

(= A falta de comunicação com o resto da Terra permitiu [ao regime] que [o regime] permanecesse mergulhado no passado)

The lack of communication with the rest of the globe has allowed to the regime to remain immersed into the past

g) reduced, infinitive prepositional clauses

Reduced, infinitive prepositions clauses, usually resulting from the reduction of relative are treated as other relatives, that is, no zero anaphor is considered (3.29):

(3.29) *Os norte-coreanos não estão sendo tratados como os iraquianos porque avalia-se que a estratégia* a ser seguida *é* `[0=indef]` *impedir que um país inimigo consiga obter armas nucleares.*

The North Koreans are not being treated as the Iraqis because it is assessed that the strategy [that is] being followed is to prohibit an enemy country from being able to obtain nuclear weapons

In this example, the NP *a estratégia a ser seguida* (the strategy being followed) is analyzed from the reduction of the relative clause *a estratégia que está sendo seguida* (the strategy that is being followed).

4. Preliminary results

In this section we present preliminary results from the annotation process of the ZAC corpus.

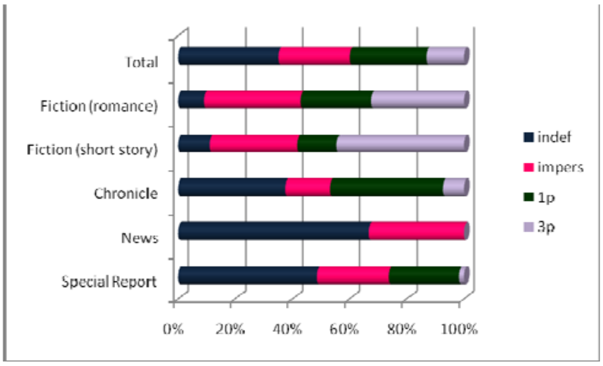
Table 2 presents the breakdown per genre of indefinite and impersonal subjects in the corpus. This type of subjects does not correspond to zero anaphors and their identification constitutes a linguistic challenge for any anaphora resolution system. Overall, they represent 401 (26.93%) from all zero subjects in the ZAC corpus.

Table 2. Indefinite/impersonal subjects per genre

ZAC corpus						
Text types	words	total marks	indef	impers	1p	3p
Special Report	15791	538	81	42	41	3
News	1769	52	8	4	0	0
Chronicle	8385	395	41	17	43	8
Fiction (short story)	3227	146	4	11	5	16
Fiction (romance)	6040	358	7	26	19	25
Total	35212	1489	141	100	108	52

Figure 1 provides a comparative overview of this subject types.

Figure 1. Indefinite/impersonal subjects per genre



The 1p and 3p indefinite zero-subject types may be targeted by using the verbal inflection as a clue and in the absence of any other candidate antecedent NP; they still represent around 10% of the corpus zeroed subjects. Indefinite zeroed subjects, without 1p or 3p inflection associated, are harder to identify. Usually with verbs in the infinitive, they represent another 10% of the zeroed subjects. Finally, the identification of impersonal constructions (around 7% cases) heavily relies on the resolution of other syntactic issues such as auxiliary constructions and temporal expressions[5].

Table 3 presents the breakdown of anaphoric and cataphoric zero anaphora per genre and also distinguishes the anaphora with intra- (< , >) and intersentencial (<< , >>) antecedent.

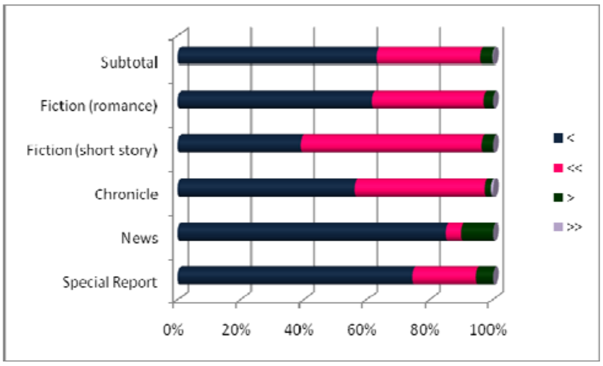
Table 3. Anaphora/cataphora breakdown per genre

ZAC corpus				
Text types	<	<<	>	>>
Special Report	275	74	20	0
News	34	2	4	0
Chronicle	156	115	5	2
Fiction (short story)	44	65	4	0
Fiction (romance)	171	99	8	0
Subtotal	680	355	41	2
Total	1035		43	

As one can see, cataphora is a relatively rare phenomenon, affecting a little over 3% of all anaphors in the corpus. Intrasentencial anaphora (<) represents 65% of all anaphors while intersentencial anaphora (<<) constitutes 34%.

There seems to be little difference among genres as far as anaphora/cataphora ratio is concerned. On the other hand distinction between intra- and intersentencial anaphors is much clearer as one can see from Figure 2. News and special reports genres show clear predominance of intrasentencial anaphora (around 80 and 70%, respectively); fiction (romance) and chronicle show average intrasentencial anaphora (around 60 and 50%, respectively); and finally fiction (short stories) only presents 40% intrasentencial anaphora. However, since the corpus is relatively small and only includes a few genres these differences may vary if a larger corpus was available and if it included other genre types.

Figure 2. Anaphora/cataphora breakdown per genre



The 23 special cases of `0(clause)` (7 cases) and `0(que)` (15 cases) represent a very rare phenomenon (1.5% of all zero subjects). The last resort '?' notation for 39 cases where a positive identification of antecedent NP is impossible represents 2.6%.

5. Future work

Based on these preliminary results we intend to develop a rule-based grammar for the identification of impersonal

[5] Since [2] do not provide explicit figures on this zero subject types it is not possible to compare our results with theirs. Nevertheless, it would be interesting to compare equivalent linguistics phenomena in both languages, Portuguese and Spanish.

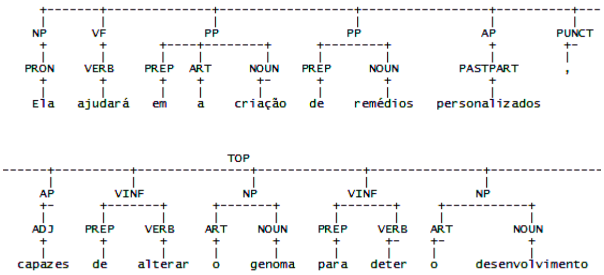
subjects that is to be integrated in the Portuguese grammar for XIP [3][4]. The temporal expressions have already been identified [7]. Auxiliary verbs involving verbs *ter* 'to have' and *haver* 'there be' are currently being implemented in XIP. It is likely that the remaining constructions of these verbs without explicit subject may be captured by rules of the XIP expressive formalism.

Secondly, we envisage a rule-base approach for the detection of the main syntactic configurations involving zero anaphors namely subordinate clauses.

Consider for example, sentence (3.19), renumbered below:

(5.1) *Ela ajudará na criação de remédios personalizados,* ***capazes*** *de* `[0=<remédios]` *alterar o genoma para* `[0=<remédios]` *deter o desenvolvimento de doenças e de transtornos psíquicos*

Figure 3. Parse tree for sentence (5.1)



This sentence contains two prepositional phrases with infinitives (*de alterar* 'of changing' and *para deter* 'for stopping'). These phrases constitute two VINF chunks (Figure 3). Since there is no NP marked with a SUBJ[ect] dependency on those verbs yet, a rule could produce with some confidence the zero anaphor.

Once the rule-based approach attains its limits, we intend to explore the machine learning techniques described by [2] and [9].

6. Acknowledgements

Research for this paper was funded by a European Union grant, under the Erasmus-Mundus Program in the International Master in Natural Language Processing and Human Language Technology. The research has been supervised by Jorge Baptista, University of the Algarve.

7. References

[1] Z. Harris. A Theory of Language and Information: A mathematical approach. Oxford: Clarendon Press, 1991.

[2] R. Mitkov. Anaphora resolution. UK:Longman, 2002.

[3] N. Mamede, J. Baptista, P. Vaz, C. Hagège. Nomenclature of chunks and dependencies in Portuguese XIP grammar (v. 2.1.). Lisboa: L2F-INESD-ID Lisboa (Internal Report), 2007.

[4] S. Ait-Mokhtar, J. Chanod, C. Roux. Robustness beyond shallowness: incremental dependency parsing. Natural Language Engineering 8 (2/3), pp. 121-144, 2002.

[5] L. Rello and I. Ilisei. A Comparative Study of Spanish Zero Pronoun Distribuition. Besançon: International Symposium on Data and Sense Mining, Machine Tanslation and Controlled Languages, pp. 209-214, 2009.

[6] S. Collovini, T. Carbonel, J. Fuchs, J. Coelho, L. Rino, R. Vieira. Summ-it: Um corpus anotado com informações discursivas visando à sumarização automática. Anais do XXVII Congresso da SBC TIL V Workshop em Tecnologia da Informação e da Linguagem Humana. Rio de Janeiro, pp. 1605-1614, 2007.

[7] C. Hagège, J. Baptista, N. Mamede. Portuguese Temporal Expressions Recognition: from TE characterization to an effective TER module implementation. 7th Brazilian Symposium in Information and Human Language Technology, SBC, 2009.

[8] M. Gross. Les bases empiriques de la notion de prédicat sémantique. Langages, 63, pp. 7-52. 1981.

[9] A. Chaves, L. Rino. The Mitkov Algorithm for Anaphora Resolution in Portuguese. A. Teixeira et al. (Eds.): PROPOR 2008, LNAI 5190, Springer-Verlag Berlin Heidelberg, pp. 51–60, 2008.

A Rule-Based Approach to the Identification of Spanish Zero Pronouns

Luz Rello* and Iustina Ilisei
Research Group in Computational Linguistics
University of Wolverhampton, Stafford Street
WV1 1SB, United Kingdom
luzrello, iustina.ilisei@gmail.com

Abstract

This paper presents a new rule-based method to identify Spanish zero pronouns. The paper describes the comparative evaluation of a baseline method for the identification of zero pronouns with an approach that supplements the baseline by adding a set of restrictions treating impersonal sentences and other zero subject expressions. The identification rules have been tested on a new corpus in which zero pronouns have been manually annotated (the Z-Corpus). The comparative evaluation shows that this rule-based method outperforms the baseline.

Keywords

zero pronoun identification; pronominal zero anaphora; subject ellipsis

1 Introduction

The identification of zero pronouns in Spanish is the first step in the development of pre-processing tools useful in NLP fields where zero anaphora resolution is necessary, *inter alia*, automatic summarisation, machine translation, question answering and the generation of multiple choice tests.

The identification and resolution of the Spanish zero pronoun is also relevant because this type of zero anaphora is fairly frequent in Spanish. Our previous research [29] on the distribution of zero pronouns reveals their ubiquity in three different genres (legal, instructional and encyclopaedic), and shows that the distribution of zero anaphors is more uniform in encyclopaedic and instructional genres than in legal texts.

This paper presents a method for the identification of zero pronouns in Spanish that distinguishes between omitted subjects which can be lexically retrieved and those that cannot. This step is essential for its resolution as the latter have to be discarded in the resolution process.

This study required the use of corpus analysis and annotation as well as deep dependency parsing techniques for the creation of a new corpus in order to test the experiments and validate the rule-based algorithm for zero pronoun identification.

Section 2 presents a description of the types of subject ellipsis that occur in Spanish, a delimitation of the zero anaphors taken into account in this research and some terminological and conceptual explanations about how zero pronouns and zero subjects have been treated throughout the previous literature, considering both linguistic and computational approaches.

The remainder of the paper is structured as follows: In Section 3 related work on the zero pronoun identification is provided while Section 4 describes the compilation and annotation of the corpora. Section 5 is devoted to the detailed description of the rule-based method which is evaluated in Section 6. Finally, in Section 7 we draw conclusions and discuss the future work.

2 Zero pronouns, zero subjects

"In spite of the widespread and fruitful use of zero signs in linguistic theory, there is no universally accepted definition of the concept of the zero linguistic sign itself. [...] Nevertheless, a maximally general definition which would cover all possible types of zero signs is a sign whose signifier is *empty*" [22].

Following this idea, two kinds of elliptic subjects are found in Spanish: implicit subjects and zero subjects. The distinction lies in the fact that while the former can be lexically retrieved (a), the latter cannot (b) [18].

(a) *zp[Vosotros]*[1] no tenéis que preocuparos.
[You] don't have to worry.

(b) Ø Llueve.
[It] is raining.

In Spanish, clauses with zero subject (b) are syntactically impersonal whereas omitted or implicit subjects (a), which are not phonetically realised, can be lexically retrieved [18].

When the phenomenon of nominal ellipsis and, specifically, the Spanish subject omission is described in the literature, both zero subjects and implicit subjects are considered cases of subject ellipsis [5]. However a consensus has emerged in which four kinds of Spanish subject ellipsis [4] are distinguished.

*This work was supported by Obra Social Fundación "la Caixa".

[1] In what follows, the sing Ø will represent zero subjects, while implicit subjects will presented between brackets and will be preceded by *zp* (zero pronoun).

1. Implicit Subject in a clause containing a finite verb[2]:

   (c) *zp[Ellos]* no vendrán.
   [They] won't come.

2. Argumental impersonal subject:

   (d) En este estudio Ø se trabaja bien.
   In this room [one] can work properly.

3. Non-argumental impersonal subject:

   (e) Ø Nieva.
   [It] is snowing.

4. Omitted subject in a non-finite verb clause:

   (f) Juan intentaba (Ø decírselo a María.)
   John tried ([John] to tell Mary.)

Moreover, elision of the subject can affect not only the noun phrase head (g), but also the entire noun phrase (h) [5].

(g) Miguel dice que los *zp[familiares, amigos...]* de María no vendrán.

Michael says that Maria's zp[friends, family...] won't come.

(h) Miguel dice que *zp[ellos]* no vendrán.

Michael says that zp[they] wont come.

Furthermore, the noun phrases affected by ellipsis in Spanish can syntactically function either as subjects (g, h) or as objects which are datives in most of the cases (i) [4].

(i) Eso *zp[nos]* induce a pensar que la noticia es falsa.

This lead zp[us] to think that the news are false.

In computational approaches to anaphora resolution [25] and specifically those focused on the resolution of pronominal zero anaphora [17], the term *zero pronoun* is used.

Some linguistic approaches make also use of the term zero pronoun which is not equivalent to the computational concept. The *Meaning Text Theory* considers a zero pronoun in subject position the same as a zero subject (Ø Llueve, *[It] is raining*) [22], while in the *Zero Hypothesis* [21] a zero pronoun can have phonetic content (full pronoun) or not (null pronoun). In this last theory, the concept of zero pronoun has to do only with its lack of lexical content in opposition to *lexical pronouns* [1].

By contrast, in NLP approaches, a zero pronoun is the zero anaphor or the resultant "gap", where pronominal zero anaphora or ellipsis occurs [25].

As in the linguistic approach, two different views about the distribution of zero pronouns have emerged. Whereas some consider that Spanish zero pronouns only appear in subject position [27], others contend that zero pronouns can occur in the object position as well [1].

As observed in our corpus, a zero pronoun can be either anaphoric (j), when it points back to its antecedent, or non-anaphoric (k), when there is no linguistic entity to which it refers. Both are examples of pronominal zero anaphora [25] and a distinction has been made between the two types during the annotation process.

(j) La costumbre$_i$ sólo regirá en defecto de ley aplicable, siempre que *zp[ella]*$_i$ no sea contraria a la moral o al orden público y que *zp[ella]*$_i$ resulte probada.[3]

The custom$_i$ will only be valid by default on the applicable law, whenever zp[it]$_i$ is not opposite to the moral a the public order and zp[it]$_i$ is passed.

(k) Los sistemas químicos que *zp[nosotros]* podemos estudiar por vía experimental son más complejos.

The chemistry systems that zp[we] can study through experimental way are more complex.

It has been noted in previous work that both phenomena, anaphoric or non-anaphoric ellipsis, are closely related [14] and occur in ambiguous clauses especially when omitting the subject (l).

(l) La serpiente estaba detrás de Pedro y sin embargo, *zp[¿él/ella?]* no se asustó [26].

The snake was behind Peter, nevertheless, zp[he or it?] was not afraid.

In conclusion, the goal of this study is to identify zero pronouns in text, *i.e.* implicit subjects —not zero subjects —which are the zero anaphors, when pronominal zero anaphora happens affecting the entire noun phrase in clauses containing finite verbs. Both types of zero pronouns, anaphoric and non-anaphoric, are taken into consideration.

3 Related work

In addition to the studies on this topic in other languages such as Japanese, Chinese, Korean or Turkish, two approaches can be distinguished in the Spanish case: either the zero pronoun identification is considered as the first step to zero anaphora resolution [17], or the identification of zero pronouns has been useful in itself in the investigation of the convergence universal [8, 9].

The most influential work on this topic is the Ferrández and Peral algorithm for zero pronoun resolution [17] together with their previous works [15, 16, 27]. In this successful method, the location of the omitted subject is based on clause identification and the detection of noun phrases which appear before the verb, unless the verb is imperative or impersonal. The mentioned authors use partial parsing while in this study deep dependency parsing and the check of whether each verb has a dependent previous or subsequent (m) subject is used.

[2] While some linguists consider these examples as a type of elision of the subject [5], it has been claimed that the subject is not elided as it is present in the verb morphology [28].

[3] Unless otherwise specified, all the examples provided from now on are taken from the Z-corpora.

(m) Se prohíben las asociaciones secretas y las de carácter paramilitar

Secret and paramilitary associations are forbidden.

The detection of verbs whose omitted subject appears before the verb in the Ferrández and Peral method [17] is very successful, however their lowest success rate (80%) corresponds with the location of verbs with zero subject. Since our set of rules are precisely focused on the treatment of this kind of verbs, they could complement this method.

Secondly, the detection of zero pronouns has been developed for the investigation of the convergence universal [8], since the number of zero pronouns in text is correlated with its degree of explicitation, which is referred to as one of the universal features of translation [3].

The method created [8] to measure the number of zero pronouns in original and translated texts is very similar to the baseline of our algorithm, although our methodology is improved by the set of constraints which reduce the false positives introduced by the baseline.

Finally, the genres under observation in this paper are different from the ones studied before, since in [17] a technical handbook (Blue Book, 15,571 words) and journalistic texts (Lexesp 9,745 words) are used, and non-annotated medical and technical texts (six corpora containing a total of 8,127,416 tokens) are employed in the studies of translation universals [9].

4 Corpora

The experiments make use of a new corpus created for this purpose named the Z-corpus. The corpus is based on selected texts from three genres: legal, instructional and encyclopaedic. It has been enlarged from its previous version [29] and some annotation errors have been corrected. The Z-corpus is under development and it is currently being annotated by a second annotator. However, this data set (with a total of 1202 zero pronouns) is relatively large compared to those used in previous work [17] whose evaluation data consisted of only 734 zero pronouns.

The legal section of the Z-corpus contains the Spanish Constitution (from the beginning up to article 110), the first book of the Civil Law Code (up to article 10 in Chapter III), the Highway Code (up to Chapter II, article 27), the Penal Code (up to Chapter II) and the Gender Equality Code (up to article 6). The instructional part is composed of four handbooks taken from the open source Wikibooks: Chemistry, Sewage Engineering, Relativity Theory, Artificial Intelligence and Time Management. Third, the encyclopaedic section is made up of 152 Wikipedia articles about mammals, medicine, linguistics and countries. Samples containing roughly equivalent numbers of zero pronouns were collected for each genre [29].

The Z-corpus was parsed with the dependency parser for Spanish Connexor's Machinese[4], which does not identify zero pronouns [7], as is the case for the rest of the Spanish parsers examined.

Each zero pronoun was annotated manually by adding an xml tag containing informative attributes. The tag was included preferably at the beginning of the clause unless this position produced an ungrammatical result. The information contained in its attributes is the following:

1. The position of the zero pronoun in the sentence.
2. The position of the antecedent of the zero pronoun.
3. The dependency head (the clause verb) on which the zero pronoun depends.
4. The kind of sentence in which the zero pronoun appears (main, subordinated, coordinated or juxtaposed).
5. The kind of clause where the zero pronoun stands (copulative, adversative, causal, relative, etc.).
6. The inter-annotator agreement score for the instance.
7. The zero pronoun is cataphoric (yes or no).
8. The antecedent of the zero pronoun corresponds with the "title" of an encyclopaedic entry or legal article (yes or no).

Point 8 turned out to be fairly significant: 168 out of 1202 zero pronouns find their antecedent in the title[5]. One example of a zero pronoun tag would be:

```
<ZERO_PRONOUN id=''w2440.5'' ant=''w2419''
depend_head=''w2441'' agreement=''high''
sentence_type=''subordinated'' title=''yes''>
```

5 Rule-based method

The algorithm for identification of Spanish zero pronouns is composed of a baseline and a set of rules. The baseline is an absolute filter which detects all the potential zero pronoun candidates (positions between tokens). The set of ordered rules are exceptions to this baseline and are applied to reduce the size of the set of candidates by discarding the false positives provided by the baseline. The candidates left over are the set of zero pronouns identified. When a zero pronoun is detected, the clause is marked as having a zero pronoun.

The objective of this study is to determine the extent to which this set of exceptions improves the baseline. The rules combine different sources of linguistic evidence: morphological, syntactic, structural and lexical. The lexical, syntactic and morphological information is provided by the parser; the semantic information is supplied by word lists which share one or more semantic features; and the structural information is contained in the rules themselves.

```
Baseline:
If a clause contains a finite verb
and it has no subject depending on this verb
there is a potential zero pronoun.
```

[4] http://www.connexor.eu/technology/machinese/.

[5] Further information about the annotation criteria and the possible values of each of the xml attributes can be found in [29].

There is a total of 27 exceptions for the baseline. These exceptions exclude potential candidates and are subsequently conditioned on a set of constraints for the sake of catching the impersonal constructions without zero pronouns included in the first set of candidates provided by the baseline.

Some studies consider different levels of Spanish impersonality (semantic and syntactic impersonality [19]) or distinguish several semantic grades in impersonality [23]. The impersonal clauses considered are the ones with zero subjects referred to in the literature as "natural impersonal clauses" [2] or "syntactic impersonal clauses" [18].

The restrictions placed upon the set of candidates catch a number of impersonal examples which are grouped in the following set of patterns:

1. Natural phenomena [6]:

   (n) Ø *Hace* mucho calor. Ø *Es* primavera. Ø *Está* nublado [13]. Ø *Llovió* cerca de 1.500ml por metro cuadrado.

   [It] is very warm. [It] is Spring. [It] is cloudy. [It] rained almost 1.500ml per square metre.

2. Temporal expressions with verbs such as "ser" [13] and "haber" [12] among others:

   (o) La película *de Ø hace* dos años.

   The film [that was] released two years ago.

3. Existential use of the verb "haber" [11] (the Spanish equivalent to *there is* and *there are*):

   (p) En un kilogramo de gas Ø *hay* tanta materia como en un kilogramo de sólido [13].

   In a kilogram of gas [there] is the same amount of mass as in a kilogram of solid.

4. Impersonal constructions with modal verbs [18, 30]. More specific examples of this rule would be "haber que" or "poder que":

   (q) Para determinar una fórmula Ø *hay que* tener en cuenta la fórmula empírica y el peso molecular.

   To determine a formula, [it] is needed to consider the empirical formula and the molecular weight.

5. Impersonal constructions with auxiliary verbs such as "ser" and "estar" [13, 18]:

   (r) Ø *Son* las dos de la tarde. [13].

   [It] is two o'clock in the afternoon.

6. Impersonal expressions with locative and the type of verbs such as "sobrar con", "bastar con" or "faltar con". [10, 31]:

   (s) Ø *Basta con* tres sesiones [13].

   [It] is enough with three sessions.

7. Pronominal unipersonal verbs with subject zero such as "tratarse de" [20, 13]. "Se" is a reflexive pronoun that detaches when the verb is conjugated:

   (t) Deberán adoptar las precauciones necesarias para su seguridad, especialmente cuando Ø *se trate de* niños.

   Necessary measures should be taken, specially when [it] concerns children.

8. Fixed constructions and idioms containing finite verbs such as "ir para" + temporal expression, "es que", "es para" or "es decir":

   (u) El peso es una fuerza, Ø *es decir*, una cantidad vectorial.

   Weight is a force, [that] is, a vectorial quantity.

9. Spanish impersonal constructions with "se" [24, 23, 32]:

   (v) Ø *Se estará* a lo que establece el apartado siguiente.

   [It] will be follow what is established in the next section.

Three rules from our system are stated below. Each rule is part of a set of rules which corresponds with patterns 1, 7 and 9 respectively.

```
for every verb with no subject do
  if [it is conjugated in third person singular] and
  [[contains the lemma "ser" or the lemma "parecer"] or
  [points at one of these lemmas]] and
  [it is followed by 0 to 3 tokens belonging to the same clause] and
  [it is followed by a member in Temporal expressions list*] then
    The clause has no zero pronoun
  end if
end for
```

* Temporal expressions list
lemma "tarde", lemma "pronto",
lemma "temprano", lemma "primavera",
lemma "verano", lemma "otoño",
lemma "invierno", lemma "enero",
lemma "febrero", lemma "marzo",
lemma "abril", lemma "mayo",
lemma "junio", lemma "julio",
lemma "agosto", lemma "septiembre",
lemma "octubre", lemma "noviembre",
lemma "diciembre", lemma "lunes",
lemma "martes", lemma "miércoles",
lemma "jueves", lemma "viernes",
lemma "sábado", lemma "domingo",
lemma "mediodía",
lemma "de" + lemma "día" in next line,
lemma "de" + lemma "noche" in next line,
lemma "de" + lemma "tarde" in next line,
text "la" + morpho "card. number" in next line,
text "las" + morpho "card. number" in next line

— RULE from set 7; unipersonal verbs:

```
for every verb with no subject do
```

```
    if [there is a there is a “se” or “Se” in text] and
    [it is followed by a verb that is conjugated in third person singular] and
    [[contains the lemma “tratar”] or
    [points at this lemma]] and
    [it is followed by 0 to 3 tokens belonging to the same clause] and
    [it is followed by text “de”] then
        The clause has no zero pronoun
    else if [there is a there is a “de” or “De” in text] and
    [it is followed by text “que se”] and
    [this is followed by a verb that is conjugated in third person singular] and
    [[contains the lemma “tratar”] or
    [points at this lemma]] then
        The clause has no zero pronoun
    end if
end for
```

Example (w) is detected by the previous rule:

(w) Ø *Se trata de* una información representacional.

[It] is about a representational information.

— RULE from set 9; Impersonal “se”:

```
for every verb with no subject do
    if [there is a token in nominative] and
    [this token contains the a “se” or “Se” in text] then
        The clause has no zero pronoun
    end if
end for
```

While (x) matches this rule, (y) does not:

(x) Ø *Se va* diciendo en la conversación.

[It] is being said during the conversation.

(y) La gramática transformacional se centra en el análisis sintáctico.

Transformational grammar is focused on syntactic analysis.

6 Evaluation

Evaluation of the identification method was carried out separately for the baseline as well as for the combination of the baseline together with the proposed rules.

It should be noted from the outset that the results are influenced by the parser's accuracy. For example, Connexor's Machinese parser does not detect every verb correctly (z):

(z) The noun “mejoras” *(improvements)* is parsed as a verb in the following:

También se introducen *mejoras* en el actual permiso de maternidad.

There are also added some improvements in the current maternity leave.

Moreover, the parser does not tag every subject in the clause (aa) successfully:

(aa) The subject is not detected in:

Las unidades de significado en la semántica léxica se denominan unidades léxicas.

The meaningful units in lexical semantics are called lexical units.

Standard evaluation measurements in the identification of zero pronouns are recall and precision rates. Evaluation was carried out for the verbs considered to have a zero pronoun. All the zero pronouns are annotated in the Z-corpus and therefore the study benefits of a gold standard for a reliable evaluation of the methods.

Baseline method applied to the Z-corpus	Precision	Recall	F-measure
Instructional	0.54	0.88	0.65
Legal	0.39	0.79	0.52
Encyclopaedic	0.39	0.74	0.50
Total	0.44	0.80	0.56

Table 1: *Baseline evaluation*

Table 1 shows the results for the baseline methodology. A higher recall rate was expected given the fact that the baseline takes into consideration a larger data set of verbs with zero pronoun candidates.

Throughout the Z-corpus the baseline has a 0.56 f-measure rate, having its highest rate for the instructional domain (0.65) with a precision of 0.54 and a recall of 0.88.

The rule-based method improves these rates to a small degree, as it is shown in Table 2. The precision and f-measure values register better results compared to the baseline, reaching up to 0.46 and 0.57 respectively. More positive results were found in the instructional genre with a precision of 0.56 and a recall rate of 0.87. However, the recall remains the same.

Rule-based method applied to the Z-corpus	Precision	Recall	F-measure
Instructional	0.56	0.87	0.67
Legal	0.40	0.79	0.53
Encyclopaedic	0.40	0.74	0.51
Total	0.46	0.80	0.57

Table 2: *Rule-based evaluation*

To sum up, the zero pronoun identification method studied in this paper has high recall and low precision. This is valuable since it is possible to filter out the unwanted cases retrieved. The rules described improve the system and may help in the identification of more general restrictions in order to obtain a better rate of precision and recall.

7 Conclusions and future work

This paper presents an improved version of a corpus annotated for zero pronouns. It has clear advantages over others of its type with regard to the number of zero pronouns annotated and the variety of genres. Additionally, the Z-corpus could be applied for the investigation of other research topics once it is be available online. The next version of Z-corpus will present an inter-annotator agreement score.

Moreover,the paper describes and evaluates a new methodology in the identification of zero pronouns. This method complements previous methodologies since a large part of our method is dedicated to the detection of impersonal structures and clauses.

Future work will be focused in overcoming parsers' errors and improving the methodology adding not only new restrictions but also preferences in order to catch the undetected examples.

In addition, a machine learning approach to tackle this topic will be applied as soon as the Z-corpus is able to offer enough training data. Furthermore, the methodology presented in this study can be used in conjunction with the findings of Spanish zero pronouns distribution [29] towards a zero pronoun resolution algorithm.

Acknowledgments

We would like to express our gratitude to Prof. Ruslan Mitkov for his supervision throughout this study. We are also especially thankful to Prof. Igor Mel'čuk for his feedback on our previous research by providing invaluable suggestions. Ana Suárez Fernández, PhD. student on Spanish syntax, pointed us important studies in the domain and helped us in the analysis of the impersonal sentences. Finally, many thanks are also due to our grammatical and mathematical conscience, Pablo Suárez García, who combed the text for errors of expression and exposition.

References

[1] L. Alonso-Ovalle and F. D'Introno. *Hispanic Linguistics at the Turn of the Myllenium*, chapter "Full and Null Pronouns in Spanish: the Zero Pronoun Hypothesis", pages 189–210. Sommerville, MA: Cascadilla Press, 2001.

[2] D. Antas. *El análisis gramatical.* Barcelona: Ediciones octaedro, 2007.

[3] M. Baker. *Text and Technology: In Honour of John Sinclair*, chapter "Corpus Linguistics and Translation Studies: Implications and Applications", pages 233–250. Amsterdam and Philadelphia: John Benjamins, 1993.

[4] J. M. Brucart. *La elisión sintáctica en español.* Barcelona: Universitat Autònoma de Barcelona, 1987.

[5] J. M. Brucart. *Gramática descriptiva de la lengua española, 2*, chapter "La elipsis", pages 2787–2863. Madrid: Espasa-Calpe, 1999.

[6] A. Calzado Roldán. La impersonalidad de los verbos metereológicos: una explicación pragmático-discursiva. *Dicenda*, 18:85–108, 2000.

[7] Connexor. *Machinese Language Model.* 1997-2006.

[8] G. Corpas Pastor. *Investigar con corpus en traducción: los retos de un nuevo paradigma.* Studien zur romanischen Sprachwissenschaft und interkulturellen Kommunikation Vol. 49. Frankfurt am Main: Peter Lang, 2008.

[9] G. Corpas Pastor, R. Mitkov, N. Afzal, and V. Pekar. Translation universals: do they exist? A corpus-based NLP study of convergence and simplification. In *8th AMTA conference*, pages 75–81, 2008.

[10] O. Fernández Soriano. On impersonal sentences in Spanish: locative and dative subjects. *Cuadernos de lingüística del Instituto Ortega y Gasset*, V:43–68, 1998.

[11] O. Fernández Soriano. Two types of impersonal sentences in Spanish: Locative and dative subjects. *Syntax, Blackwell Publishing*, 2-2:101–140, August 1999.

[12] O. Fernández Soriano and G. Rigau i Oliver. Construcciones temporales no impersonales en español. In *Estudios de lingüística del Español (ELiEs), Actas del II Congreso de la Región Noroeste de Europa de la Asociación de lingüística y Filología de América Latina (ALFAL)*, volume 22, 2005.

[13] O. Fernández Soriano and S. Táboas Baylín. *Gramática descriptiva de la lengua española, 2*, chapter "Construcciones impersonales no reflejas", pages 1631–1722. Madrid: Espasa-Calpe, 1999.

[14] A. Ferrández, A. Palomar, and L. Moreno. El problema del núcleo del sintagma nominal: ¿elipsis o anáfora? *Procesamiento del lenguaje natural*, 20:13–26, 1997.

[15] A. Ferrández, A. Palomar, and L. Moreno. Anaphor resolution in unrestricted texts with partial parsing. In *Annual Meeting of the ACL, Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 385–391, 1998.

[16] A. Ferrández, A. Palomar, and L. Moreno. An empirical approach to Spanish anaphora resolution. *Machine Translation*, 14-3/4:191–216, 1999.

[17] A. Ferrández and J. Peral. A computational approach to zero-pronouns in Spanish. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL2000)*, pages 166–172, 2000.

[18] L. Gómez Torrego. *La impersonalidad gramatical: descripción y norma.* Arco Libros, Madrid, 1992.

[19] L. Gómez Torrego. *La enseñanza de la lengua en la década de los noventa*, chapter "Impersonalidades, generalizaciones, encubrimientos y metonimias en la gramática", pages 37–58. Madrid: Universidad Autónoma de Madrid, 1994.

[20] L. Hernando Cuadrado. Sobre la expresión de la impersonalidad. In *ASELE. Actas IV*, Marrakech, Morocco, 1994. Asociación de Profesores de Español como Lengua Extranjera (ASELE).

[21] A. Kratzer. More structural analogies between pronouns and tenses. In *The Proceedings of Semantics and Linguistic Theory VIII (SALT VIII).* Cornell University: CLC Publications, 1998.

[22] I. Mel'čuk. *Aspects of the Theory of Morphology.* Mouton the Gruyter, Berlin, New York, 2006.

[23] A. Mendikoetxea. *Las Construcciones con "se"*, chapter "La Semántica de la Impersonalidad,", pages 239–267. Madrid: Visor, 1994.

[24] A. Mendikoetxea. *Gramática descriptiva de la lengua española, 2*, chapter "Construcciones con *se*: Medias, pasivas e impersonales", pages 1575–1630. Madrid: Espasa-Calpe, 1999.

[25] R. Mitkov. *Anaphora resolution.* London: Longman, 2002.

[26] J. C. Moreno Cabrera. Tipología de la catáfora paratáctica: entre la sintaxis del discurso y la sintaxis de la oración. *Estudios de lingüística*, 3:165–192, 1985.

[27] J. Peral and A. Ferrández. *Lecture Notes In Computer Science*, volume 1835, chapter "Generation of Spanish Zero-Pronouns into English", pages 252–260. London, UK: Springer-Verlag, 2000.

[28] RAE. *Esbozo de una nueva gramática de la lengua española.* Madrid: Espasa Calpe, 1977.

[29] L. Rello and I. Illisei. A comparative study of Spanish zero pronoun distribution. In *Bulag 33: International Symposium on Data and Sense Mining Machine Translation and Controlled Languages, and their application to emergencies and safety critical domains*, pages 209–214. Presses universitaires de Franche-Comté (Presses-ufc), July 2009.

[30] G. Rigau i Oliver. Los predicados impersonales relativos en las lenguas románicas. *Revista española de lingüística*, 29-2:317–356, 1999.

[31] M. L. Rivero. Spanish quirky subjects, person restrictions, and the person-case constraint. *Linguistic Inquiry, Massachusetts Institute of Technology*, 2004.

[32] M. Suñer. Demythologizing the impersonal "se" in Spanish. *Hispania*, 59-2:268–275, 1976.

Exploring Context Variation and Lexicon Coverage in Projection-based Approach for Term Translation

Raphaël Rubino
Laboratoire Informatique d'Avignon
339, chemin des Meinajaries
Agroparc BP 1228
84911 Avignon Cedex 9, France
raphael.rubino@univ-avignon.fr

Abstract

Identifying translations in comparable corpora has inspired many studies in bilingual terminology extraction [4, 5]. Projection-based approaches, which are among the most popular ones, rely on a seed bilingual lexicon. Surprisingly, there is no careful analysis of the impact of the size the initial context and coverage of the lexicon. This is precisely the focus of this study. We observe that source context size and lexicon coverage influence robustness in projection-based term translation. In particular, we show that increasing the number of seed words by a factor of three leads to a 20% relative improvement in accuracy.

1 Introduction

Parallel corpora have been widely used in machine translation. For example, sentence and word aligment models proposed by [2], or applications to multilingual terminology extraction [8]. This approach, using parallel corpora, yields good results. But the lack of parallel texts is still an issue. This is particularly true for specific domains. Building parallel corpora is time-consuming and relies heavily on human translators. Even if domain specific parallel corpora exist, terminology extraction often amounts to reverse engineering the work of human translators.

This is a reason why comparable corpora [11] are studied by researchers in multilingual terminology extraction. Some authors have shown that statistical methods can make use of comparable corpora. In particular, [17] paved the way for a family of approaches that assumes that co-occurrences of words which are translations of each other are correlated in comparable corpora. Basically, we can observe that a word and its translation appear in the same lexical environment, which can be used as a context vector [5]. This projection-based translation approach can be applied to terminology extraction. For example in [14], a term extraction program coupled with a lexical alignment program are implemented to manage this task.

Usually, in projection-based term extraction, a bilingual or multilingual lexicon is needed to do the projection from one language to another. This step depends on the lexicon and the context vector. To the best of our knowledge, there are no analyses on the impact of the initial context size or on the coverage of the lexicon in such projection-based methods. This is precisely the focus of this paper. The remainder of this paper is organized as follows : in section two, we present the projection-based approach and describe related work. In section three, we explain the experimental settings. In section four, we present the resources used. Finally the results are presented in section five, followed by a discussion in section six.

2 Projection-based Approach

2.1 Description

In the source language text, the term to be translated is surrounded by a context consisting of other terms. This information helps us build a context vector, with a flexible window around the term [6, 14] for example. Then this context has to be projected in the target language. The target context vector is built thanks to a bilingual lexicon. This lexicon-based step is the basis of projection-based approaches. To retrieve translation candidates, the projected context vector has to be compared with all possible vectors in the target language built directly from corpora.

This comparison can be computed with different similarity measures. Usually, the Cosine, Jaccard or Dice coefficient [9] are used. [18] obtained better results using the city-block metric than using the Cosine, Jaccard coefficient, Euclidean distance and scalar product. [14] have also made studies on the impact of different metrics to extract terms' translation pairs. The figure 1 illustrates this projection-based approach.

2.2 Related Works

Studies on parallel corpora has allowed to identify features like co-occurrence position of a word and its translation [10, 19, 21]. Switching to non-parallel corpora implies that the co-occurrence feature is not directly applicable because there are no direct correspondences between sentences or segments. Another word feature correlating words pairs, called context heterogeneity [5], can be applied to texts in different languages which are not translations of each other. Computed on comparable corpora, the context heterogeneity measure can be used to retrieve domain-

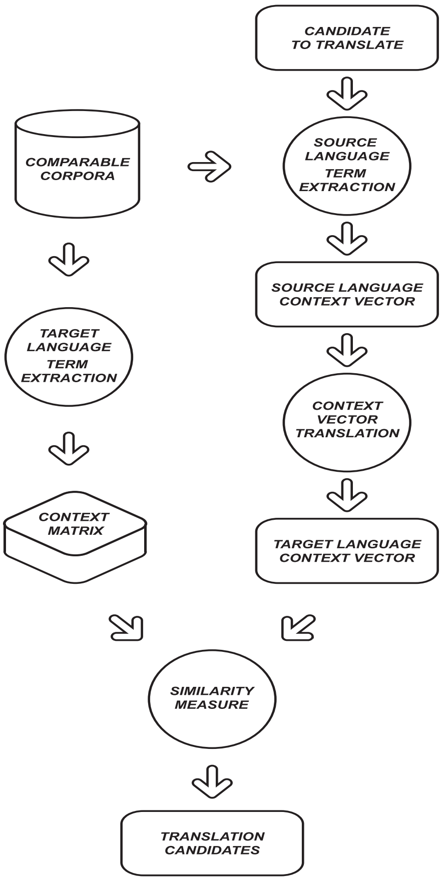

CANDIDATE TO TRANSLATE
COMPARABLE CORPORA
SOURCE LANGUAGE TERM EXTRACTION
SOURCE LANGUAGE CONTEXT VECTOR
TARGET LANGUAGE TERM EXTRACTION
CONTEXT VECTOR TRANSLATION
CONTEXT MATRIX
TARGET LANGUAGE CONTEXT VECTOR
SIMILARITY MEASURE
TRANSLATION CANDIDATES

Fig. 1: *General representation of the projection-based approach with comparable corpora.*

specific word pairs in unrelated languages (English and Japanese for example).

[17] concludes that patterns of co-occurrences of words between different languages are correlated in non-parallel corpora. In [18], co-occurrence matrices are built from comparable corpora, and used to compare the projected vectors with all possible vectors from the initial text. In the same study, a small seed word lexicon, which does not cover the test set, is used and expanded during the experiments with the projection-based approach.

Based on these studies, [14] proposed an approach to solve multi-word term translation from non-parallel corpora. They first adapt the single-word term context vector approach proposed by [7] to multi-word term. Then, an implementation of the direct context vector method is proposed and applied to terminology extraction between unrelated languages (French/Japanese). Different metrics are compared to compute similarity between the context vector of the term to translate and the context-matrix built from the initial corpora [14].

In these studies, much attention is paid to similarity metrics between context vectors, built on single or multi-word term co-occurrence values. The projection-based approach described here requires a bilingual seed word lexicon, and it is very surprising that its coverage is not yet carefully studied. Only terms in the lexicon are projected in the target context vector, so this aspect is very important for initial context-vector projection.

3 Experimental Settings

Our system takes as input a list of domain-specific single and multi-word terms to translate. Each of them is a query for document retrieval in a comparable corpus. Terms from these documents compose an initial context vector. Then we use a seed word lexicon to project the context from source to target language. The resulting target language vector is used to retrieve documents from a comparable corpus. The aim of this study is to manage and score document retrievals containing initial term translations, according to initial translation references. The impact of the initial context's size variations, its seed word lexicon coverage and the lexicon size are studied. Experiments were conducted on French to English single and multi-word term translations. The implementation is composed of five parts :

1. Document retrieval (the query is the term to translate)

2. Initial context vector construction from words contained in documents

3. Projection of the vector in the target language using the seed word lexicon

4. Document retrieval with projected vector

5. Oracle scoring on documents : containing or not the term translation

The score is computed on retrieved documents. If a returned document contains the candidate translation, an oracle is set to 1, otherwise it remains at 0. It is possible that the initial context vector can not be built, because the corpus does not contain the term to translate. It is also possible that the projected vector cannot be used to retrieve documents in the target language. The first reason is that the translation reference is not in the corpora. We decide then to compute two oracle scores : on all candidates from the initial term list and on candidates with an initial context covered by the corpora.

The task is to verify whether the target language context is robust enough to retrieve documents containing the initial term translation. It is the first thing we want to measure. We make variations of the initial context and the seed word lexicon size, but also of the seed word lexicon coverage.

4 Resources

4.1 Comparable Corpora

Using the World Wide Web as a non-parallel corpus can solve the problems of accessibility, relevance and quantity of data. Wikipedia is a well known online free collaborative encyclopedia. Many articles are domain specific, and each document represents one concept only [13]. As in [16], we use Wikipedia[1] as a comparable corpus, for the abundance of the multilingual content freely available. Wikipedia is used in many natural language processing domains, like named entity disambiguation [3], the retrieval of similar sentences across different languages [1], thesaurus extraction [13], semantic relation extraction [20], etc.

Although Wikipedia has a structure that can help identify translations (cross languages links, titles of pages and section, ...), we do not consider this information this study. We want to build the context vectors for terms to translate with Wikipedia articles, which are used like concept-related word lists or semantic networks.

In order to extract the information we need from Wikipedia, we rely in this work on a tool called NLGbAse[2]. This tool provides a search engine with cosine similarity computed between the query and the returned documents. For each document, NLGbAse gives the list of contained words ranked by their *tf.idf* measure computed on the whole Wikipedia corpora.

4.2 Candidates

In this series of experiments, a term list is taken from the MeSH thesaurus [15]. 10 000 single and multi-words terms in French, along with their translations in English, are extracted [12]. None of these terms are covered by the lexicon used for the experiments. Prior to the experiments, two filtering steps are done. The first is made to be sure that Wikipedia can be used to build source language contexts. This means that the term to translate is in the corpus. Then, a second filtering step is done on Wikipedia with the translation proposed in MeSH for every term. After these two steps, 2 000 terms were removed, because none of them are covered by Wikipedia.

4.3 Bilingual Seed Word Lexicon

To manage a pivot between different languages, a domain specific lexicon has to be built. We use the data available from Robert H. Vander Stichele's website[3]. We automatically retrieve a French-English lemma collection of technical and popular medical single words. To handle general terms, we choose to extend the 1800 medical words collection with a general lexicon containing 3200 words. Our lexicon finally contains 5000 word pairs.

For the experiments, three seed word lexicons are used. The first is the full one, with general and domain specific words. The second is only general, and the third is only medical.

We choose to automatically build lexicons from a web resource to be as independent as possible from the candidates to translate extracted from the MeSH thesaurus. The lexicon coverage of the candidate list is null, in order to avoid any bias. It means that only words in context vectors are translated.

4.4 Stop-words List

Words used to build initial context-vectors are more or less significant and can even introduce noise. For example, words in the source language which are not directly related to the term to translate can be "ainsi" ("so"), "quand" ("when") or "toujours" ("always"), etc. We decided to filter these words before building context-vectors with a stop-words list, which contains the 1300 common non-content words of the source language.

5 Results

Table 1 presents the results of the oracle described in the experimental settings. We give details about the number of terms to translate and about the number of seed words (corresponding to the size of the projected context). We also study how the number of documents and the number of terms per document vary. We use the full seed word lexicon (general and domain specific).

The maximum number of candidates handled during the experiments does not reach the maximum number of candidates in the initial list. This can be explained by the lack of vocabulary in the seed word lexicon. A null initial context-vector cannot be used to do a projection-based approach, so the candidate to translate is not handled.

docs	terms	cand.(%)	seeds	oracle	limited
1	10	63.25	2.06	0.29	0.58
1	50	89.54	6.37	0.45	0.64
1	100	94.52	11.07	0.49	0.67
1	200	94.86	19.95	0.53	0.72
1	999	94.90	38.69	0.56	0.76
10	1	49.51	1.74	0.21	0.54
10	2	65.62	2.50	0.29	0.57
10	5	82.30	4.58	0.40	0.63
10	10	89.24	7.99	0.48	0.69
10	20	92.24	14.78	0.53	0.75
10	50	94.00	32.45	0.58	0.80
10	100	95.00	58.16	0.61	0.83
10	200	95.00	103.77	0.63	0.85
20	1	57.91	2.28	0.25	0.56
20	5	84.41	6.81	0.44	0.67
20	10	89.92	12.09	0.51	0.72
20	20	92.38	22.29	0.55	0.77
20	40	93.82	40.10	0.59	0.80
20	50	94.00	48.50	0.60	0.81
20	100	95.00	86.09	0.62	0.84
50	1	64.82	3.31	0.30	0.59
50	2	75.52	5.45	0.37	0.63
50	5	85.20	11.06	0.47	0.70
50	10	90.11	19.66	0.52	0.75
50	20	92.44	35.75	0.57	0.79
100	1	67.38	4.35	0.32	0.61
100	10	90.18	26.93	0.53	0.75

Table 1: *Oracle with the full seed word lexicon. Initial documents and terms are used to build the initial context. The last column is an oracle computed on covered candidates only (third column). The "oracle" column is computed on all candidates (8000 term pairs).*

[1] http://www.wikipedia.org
[2] http://nlgbase.org/
[3] http://users.ugent.be/∼rvdstich/eugloss/welcome.html

In order to reach an accuracy of 50%, 20 initial documents and 10 words per document are needed. The source context-vector is then sufficient to do the projection in the target language and to retrieve good documents. In theory, this means that an initial context-vector of 200 terms is required. Besides, the average projected context-vector size is about 12.09 seeds. The seed word lexicon coverage of the initial context is, in theory, about 6.04%. In fact, the size of the initial context-vector depends on the number of terms contained in the documents from Wikipedia. The average lexicon coverage of initial contexts in all experiments is around 11% (from 9% to 13% depending on the size of the initial context-vectors).

On the experiments with 20 initial documents, we can see that increasing the number of seeds by a factor of three leads to a relative improvement in oracle accuracy of 20%. If we look at the results on 1 document and 50 terms per document, compared to the results on 20 documents with 5 terms, for a lower number of seeds (6.37 instead of 6.81) and half of the initial context-vector size (50 words instead of 100), the oracle accuracy of document retrieval is enhanced. It can be explained by a higher seed word lexicon coverage, associated to a *better* initial context-vector. In fact, this context-vector is closer to the candidate to translate. All the words used to build the context are in the first document retrieved by the search engine, that is the document with the best cosine similarity measure.

The tables 2 and 3 contain the oracle and *limited* scores, respectively with the general and the medical seed word lexicons. We can see that a higher recall is obtained with the domain-specific lexicon. The reason is that this seed word lexicon gives a higher initial context coverage. Using the general seed word lexicon, we see that more term-to-translate candidates are handled. This means that it is easier to project contexts, but the recall (documents with good translation retrieval) is lower than with domain specific lexicon.

docs	**terms**	**cand.(%)**	**seeds**	**oracle**	**limited**
1	10	28.61	1.36	0.09	0.42
1	50	76.68	2.64	0.21	0.36
1	100	91.67	4.71	0.26	0.37
10	1	19.80	1.21	0.07	0.44
10	10	76.90	3.28	0.26	0.43
10	20	85.94	5.65	0.32	0.48
10	50	91.88	12.69	0.39	0.54
10	100	94.71	25.33	0.43	0.58
10	200	95.00	50.03	0.46	0.62
20	5	69.03	3.02	0.24	0.44
20	20	86.39	8.94	0.35	0.52
20	40	90.45	16.41	0.40	0.57
50	1	41.19	1.76	0.14	0.43
50	10	79.78	8.97	0.33	0.54
50	50	91.96	34.51	0.43	0.60
50	100	94.71	61.71	0.46	0.62

Table 2: *Oracle with the general seed word lexicon.*

On figure 2, the oracle score by seed word is reported. We can see that for an identical number of seeds, the number of documents used to build the initial context-vector has an important impact. Taking less documents but more terms by document increases the oracle score, with the same seed word lexicon. On figure 3, the impact of the type of seed word lexicon is presented. We can see that using only a domain specific seed lexicon, instead of using only a general lexicon, leads to an improvement of the oracle accuracy.

docs	**terms**	**cand.(%)**	**seeds**	**oracle**	**limited**
1	10	45.18	1.99	0.19	0.55
1	50	79.52	5.20	0.36	0.58
1	100	86.99	8.22	0.41	0.61
10	1	35.67	1.65	0.15	0.53
10	10	80.36	6.42	0.39	0.63
10	20	88.07	11.02	0.45	0.66
10	50	92.73	22.61	0.52	0.72
10	100	93.53	38.22	0.56	0.76
10	200	94.73	62.21	0.59	0.79
20	5	74.38	5.52	0.35	0.61
20	20	89.73	15.79	0.48	0.69
20	40	92.59	27.23	0.53	0.73
50	1	50.59	2.92	0.22	0.55
50	10	86.02	13.55	0.45	0.67
50	50	93.01	48.2	0.56	0.77
50	100	93.56	77.86	0.59	0.80

Table 3: *Oracle with the domain specific seed word lexicon.*

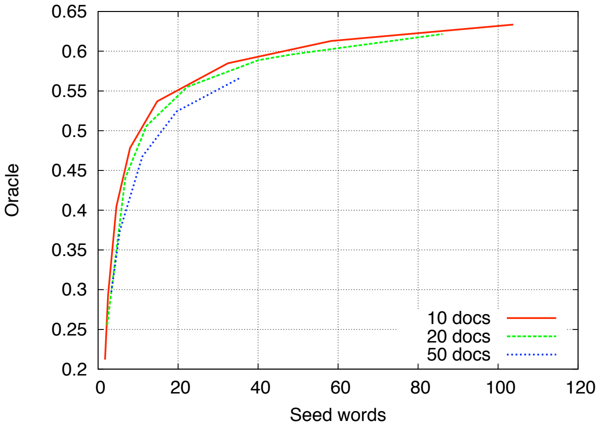


Fig. 2: *Oracle score with seed word variations.*

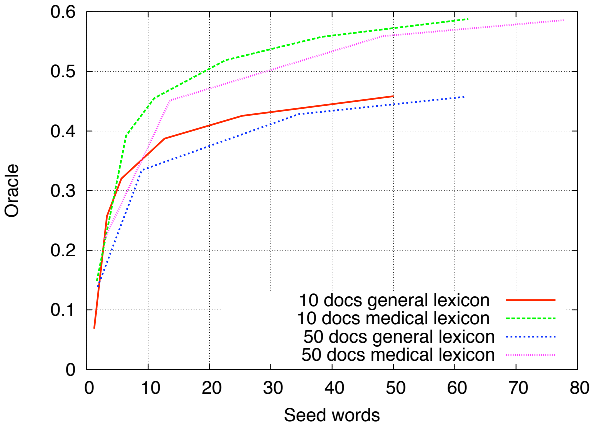


Fig. 3: *Oracle score by lexicon type.*

6 Discussion

In this paper, we describe the impact of the initial context-vector size and its lexicon coverage in projection-based methods for term translation. We also give details about the robustness of this context with variations of the number of documents and term-by-document during its construction. The oracle accuracy can be improved with less documents initially used and a higher seed word lexicon coverage. With an equivalent number of seeds, the smaller the initial context-vector, the higher the oracle accuracy.

We show that for domain specific term translation, a recall score of 85% can be obtained with a domain specific seed word lexicon, completed with a general lexicon. The domain specific lexicon has a better coverage of the initial context while the general lexicon handles more term-to-translate candidates.

This study will help us to continue our works on projection-based domain specific term translations. In particular, using other comparable corpora, like the World Wide Web [7], which can be considered as an higher unrelated non-parallel corpora. We assume that generation of N-Best translation candidates lists, which is a step further in this study, can be improved with robust initial context-vectors.

References

[1] S. Adafre and M. de Rijke. Finding Similar Sentences Across Multiple Languages in Wikipedia. In *Proceedings of the 11th EACL conference*, pages 62–69, 2006.

[2] P. Brown, S. Della Pietra, V. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer, and P. Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, 1990.

[3] S. Cucerzan. Large-scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of EMNLP-CoNLL conference*, pages 708–716, 2007.

[4] B. Daille, É. Gaussier, and J. Langé. Towards Automatic Extraction of Monolingual and Bilingual Terminology. In *Proceedings of the 15th COLING conference*, volume 1, pages 515–521. ACL, 1994.

[5] P. Fung. Compiling Bilingual Lexicon Entries from a Non-parallel English-Chinese Corpus. In *Proceedings of the 3rd Workshop on Very Large Corpora*, pages 173–183, 1995.

[6] P. Fung. A Statistical View on Bilingual Lexicon Extraction: from Parallel Corpora to Non-parallel Corpora. *Lecture Notes in Computer Science*, 1529:1–17, 1998.

[7] P. Fung and L. Yee. An IR Approach for Translating New Words from Nonparallel, Comparable Texts. In *Proceedings of the 36th ACL conference*, pages 414–420. ACL, 1998.

[8] É. Gaussier. Flow Network Models for Word Alignment and Terminology Extraction from Bilingual Corpora. In *Proceedings of the 36th ACL conference*, pages 444–450. ACL, 1998.

[9] W. Jones and G. Furnas. Pictures of Relevance: A Geometric Analysis of Similarity Measures. *Journal of the American society for information science*, 38(6), 1987.

[10] J. Kupiec. An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora. In *Proceedings of the 31st ACL conference*, pages 17–22. ACL, 1993.

[11] J. Laffling. On Constructing a Transfer Dictionary for Man and Machine. *Target*, 4(1):17–31, 1992.

[12] P. Langlais, F. Yvon, and P. Zweigenbaum. Translating medical words by analogy. In *Intelligent Data Analysis in Biomedicine and Pharmacology*, pages 51–56, Washington, DC, USA, 2008.

[13] D. Milne, O. Medelyan, and I. Witten. Mining Domain-specific Thesauri from Wikipedia: A Case Study. In *IEEE/WIC/ACM International Conference on Web Intelligence, 2006. WI 2006*, pages 442–448, 2006.

[14] E. Morin, B. Daille, K. Takeuchi, and K. Kageura. Bilingual Terminology Mining-Using Brain, not Brawn Comparable Corpora. In *Proceedings of the 45th ACL conference*, page 664. ACL, 2007.

[15] S. Nelson and J. Schulman. A Multilingual Vocabulary Project-Managing the Maintenance Environment. *MeSH Section, National Library of Medicine, Bethesda, Maryland*, 2007.

[16] M. Potthast, B. Stein, and M. Anderka. A Wikipedia-based Multilingual Retrieval Model. *Lecture Notes in Computer Science*, 4956:522, 2008.

[17] R. Rapp. Identifying Word Translations in Non-parallel Texts. In *Proceedings of the 33rd ACL conference*, pages 320–322. ACL, 1995.

[18] R. Rapp. Automatic Identification of Word Translations from Unrelated English and German Corpora. In *Proceedings of the 37th ACL conference*, pages 519–526. ACL, 1999.

[19] F. Smadja and K. McKeown. Translating Collocations for Use in Bilingual Lexicons. In *Proceedings of the ARPA HLT*, volume 94, 1994.

[20] M. Strube and S. Ponzetto. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In *Proceedings of the AAAI conference*, page 1419. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[21] D. Wu and X. Xia. Learning an English-Chinese lexicon from a Parallel Corpus. In *Proceedings of the 1st AMTA*, pages 206–213. AMTA, 1994.

An Evaluation of Output Quality of Machine Translation Program

Mitra Shahahbi
MA. Student
University of Wolverhampton
Stafford Street
Wolverhampton WV1 1NA
United Kingdom
Shahabi_mitra@yahoo.com

ABSTRACT

This article reports an exploratory evaluation of the output quality of two prevalent English-Persian Machine Translation programs. The purpose of the research is to find out which program produces relatively better output, and what major linguistic bottlenecks MT programs will encounter in their processing of texts. Criteria were established in light of structural theories to solve the MT output from the perspective of Accuracy and Intelligibility. For each program, the mean score it obtained for its output and the rate of correctness of its translation of the testing points were calculated. An analysis of the mean score and the rate of correctness of each program generated the following findings about the output quality of these two programs: 1) Padideh Translator produces the best output. 2) The major linguistic bottlenecks in English-Persian MT programs occur in the areas of morphology, complex sentences, syntactic ambiguity and semantic analysis, generation of Persian, and long sentences.

KEYWORDS

Machine Translation, Accuracy, Intelligibility

1. INTRODUCTION

MT technology is very important in the future of business. More and more business is being done on the Internet. People from every country are starting to consider the World Wide Web a mall from which they can buy anything they need [3]. Although English is spoken widely across world; it is the 4th most spoken language, and is by far the most extensively used language to communicate science, propagate technology and do business, not all potential users have access to this language This leaves many potential customers who do not understand the English-only websites on the internet. MT helps the business adapt to the customers.

The economic necessity of finding a cheaper solution to international exchange has resulted in continuing technological progress in terms of translation tools designed to automate and computerize the translation of natural language texts or to use computers as an aid to translation [2].

Although MT has some disadvantages, we will be able to use MT for cheaper and faster translation in near future. The present, relatively poor quality of translation yield by the computer: where total grammatical, semantic/associative meanings and pragmatic adequacy are concerned, it can lead the native speaker to reject the text on the grounds that it is strange, awkward, and even nonsensical [1] when reading. But there are also good reasons why we use machines to translate our texts. The primary reasons for using machine translation are speed, cost savings, and availability. John Hutchins [8] summarizes the reasons for using computers in translation as follows and insists any one of these may justify MT or computer aids:

- Too much translation for humans
- Technical materials too boring for humans
- Greater consistency required
- Need results more quickly
- Not everything needs to be top quality
- Reduce costs

MT prompts researchers to ask whether it is possible that we have MT systems that can produce translation that is as good as human translation but faster and cheaper. What programs produce relatively better translation? And what difficulties are most MT programs confronted with? This article intends to probe into these questions and report an exploratory evaluation of the output quality of two prevalent English-Persian MT programs, namely *Pars Translator* and *Padideh Translator*.

1.1. Machine Translation Evaluation (MTE)1

The general agreement about the basic features of MT evaluation are not, at the outset, subject to much dissention, but there are no collectively acknowledged and reliable methods and measures, and evaluation methodology has been the subject of much discussion in recent years.

"As in other areas of NLP [5], three types of evaluation are recognized: **Adequacy Evaluation** to determine the fitness of MT systems within a specified operational context; **Diagnostic Evaluation** to identify limitations, errors and deficiencies; and **Performance Evaluation** to assess stages of system development or different technical implementations. Adequacy evaluation is typically performed by potential users and/or purchasers of system; diagnostic evaluation is the concern mainly of researchers and developers; and performance evaluation may be undertaken by researchers, developers, or potential users.

MT evaluations typically include features not present in evaluations of other NLP systems. The quality of the raw translations, e.g., intelligibility, accuracy, appropriateness of style/register; the usability of facilities for creating and updating dictionaries, for post-editing texts, for controlling input language, for customization of documents, etc.; the extendibility to new language pairs and/or new subject domains; and cost-benefit comparisons with human translation performance."

According to Hutchins and Somers [4] the most obvious tests of the quality of a translation are:

A. **Accuracy**, that is the extent to which the translation accurately renders the meaning of the source text, without intensifying or weakening any part of the meaning [7]; and
B. **Transparency**, which is the extent to which the translation appears to a native speaker of the TL to have originally been written in that language, and conforms to the language's grammatical, syntactic and idiomatic conventions [7].

The evaluation made in this research focused on the quality of the output, i.e., the translation of two prevalent English-Persian MT programs.

2. METHODOLOGY OF THE STUDY

Two prevalent English-Persian MT programs (Pars Translator & Padideh Translator) were selected as the subject of this research. The instruments of the research included a computer, a test suite[2] and detailed criteria for measuring the output.

The criteria for selecting these MT programs were:

a) Commercially available in Iran or accessible on the Internet
b) Presently and popularly used
c) Fully-automated

Given that the sentence is the basic unit in the translation process of the two programs evaluated, it was chosen as the basic testing item. Hence this research only evaluated the output quality of sentences.

Altogether there were 451 sentences, containing 282 testing points and covering 9 subjects, namely lexical coverage, phrase, morphology, simple sentences, complex sentences, syntactic ambiguity and semantic analysis, generation of Persian, special difficulties in English-Persian machine translation, and long sentences.

The test suite borrowed from the language discrete-point testing method, which means each testing item (i.e., sentence) in the suite contains a testing point. It consists of 3 parts:

- ❖ Testing an MT system' ability to analyze SL (from subject 1 to subject 6)
- ❖ Testing an MT system's ability to synthesize TL (subject 7)
- ❖ Testing an MT system's ability to deal with special difficulties in MT (subject 8 and subject 9)

The test suit I employed in the present research was originally in Chinese. The present English version was translated from Chinese by Yan Weiwei [9] served as a useful and instructive model and fitted well in with the research.

According to what is concerned with measuring accuracy, the focus was on the preservation of meaning, which involves the comparison of meaning in the output with that in the original.

Ke ping [6] distinguishes three kinds of meaning in translation in a socio semiotic approach, based on which (excluding those meanings that are impossible in or irrelevant to MT) accuracy was measured in two

[1] It is the evaluation of an MT program. There is no simple or unique way of conducting an MTE.

[2] In Natural Language Processing (including Machine Translation), a test suite is a set of points, artificially constructed and designed to probe the system's behavior with respect to some particular language phenomena. [5]

dimensions: **referential meaning** (at lexical level, the lexical meanings of the words and phrases) and **grammatical meaning** (inflectional morphology and syntax)**.** Grammatical categories include tense, aspect, case, gender, mood, number, person, and voice.

In this research the measure of intelligibility revolved around two dimensions, i.e., grammaticality and fluency.

The scoring procedure embraced two types: The first step was to decide whether the meaning of a translation is completely unfaithful to that of the original or totally unintelligible. In either case, the translation will score zero.

In the second step, those translations not having scored zero, were assessed a set of scoring criteria as follows:

The full mark of each translation was 10 points: 5 for accuracy and 5 for intelligibility. Different weights were assigned to referential meaning, grammatical meaning, grammaticality, and fluency. Every translation lost points according to the nature and number of the errors it made. For an error at 'referential meaning at sentence level', 1.0 point was missing. Once an error was made at the level of syntax, 0.5 point was subtracted. The same weight was also given to 'word order', collocation, and idiomaticity. For those below the sentence level, i.e., lexical meaning errors, wrong tenses, etc, 0.25 was subtracted. Punctuation also lost 0.25 point. After all the points caused by all the errors were subtracted from the 10 points, the remaining points were the final score that reflected the overall quality of the output.

All the sentences in the suite were translated on computer by two MT programs. The output was gathered for further analysis. Then the criteria for measurement were applied to score the output of each program, and whether the programs had correctly translated each testing points.

The average of the total scores and the average of each subject were calculated. Then the overall rates of correction of the translation of the testing points and the rates of the correctness on each subject were also calculated. The average of the total scores and the overall rates of correctness were compared to answer the first question of the research. The rates and the means on each subject were compared and the translation of the testing points was analyzed to answer the second research question.

3. RESULTS & DISSCUSSION

The findings about the output quality of these two programs are as follows:

a) It was reflected that Padideh Translator scored a higher rate of correctness (with a slight difference of 0.5%).

b) The major linguistic bottlenecks of these MT programs in translating testing points were related to Morphology, Complex Sentences, Syntactic Ambiguity & Semantic Analysis, Generation of Persian, and Long Sentences.

3.1. Morphology

In Pars:

- Infinitives as adverbials
- The passive voice and different tenses of the Infinitive
- The perfect form and the passive voice of present participle

In Padideh:

- Special usages of comparative degree of adjectives
- The addition of *–est* to form the superlative degree of some adverbials
- The addition of preceding *most* to form superlative degree of some adverbials
- Infinitive as subject
- The perfect form and the passive voice of a gerund

Shared in Padideh and Pars:

- The addition of a preceding *more* to form the comparative degree of adverbials
- The base form, the past form, and the past participle of a verb are the same
- The meaning and translation of structure *too+ adjective/adverb+ infinitive*
- Present participle phrases as attributive placed behind the noun it modifies
- Present participle as adverbial denoting time, result, reason, condition and purpose (with the same function as that of a sentence or clause)

- Gerund as attributive
- The complex construction of gerund as subject, object, prepositional object and predicative

3.2. Complex Sentences

In Pars:

- Subject clause introduced by *what*
- Subject clause introduced by *who* or *whoever*
- Adverbial clause of reason
- Adverbial clause of manner
- Predicative clause introduced by *why*

In Padideh:

- Subject clause introduced by *when*
- Predicative clause introduced by *who* or *whom*
- Adverbial clause of comparison
- Attributive clause introduced by *preposition + which* or *preposition + whom* construction
- Appositive clause

Shared in Padideh and Pars:

- Attributive clause introduced by *which, who, whose* or *whom*
- Subject clause introduced by *where* or *wherever*
- The attributive clause is also a complex sentence.
- Neither of the two clauses in one sentence is embedded in the other clause.

3.3. Syntactic Ambiguity & Semantic Analysis

In Pars:

- Word belonging to adjective and verb
- Complex sentences which are semantically compound ones

In Padideh:

- Word belonging to adverbial and adjective
- Word belonging to adverbial and preposition
- Word belonging to noun, adjective and verb
- Word belonging to conjunction, adjective and verb
- Word belonging to demonstrative pronoun, relative pronoun and subordinate conjunction
- Nouns with different meanings
- Subordinate conjunctions which have various meanings and introduce different types of clauses

Shared in Padideh and Pars:

- Word belonging to pronoun and relative pronoun
- Word belonging to conjunction and preposition
- Word belonging to conjunction and adverb
- Verbs with different meanings in accordance with what follows
- To judge whether the present participle helps to form the predicate verb or act as a nominal modifiers

3.4. Generation of Persian

In Pars:

- The definite articles are often omitted.
- The translation of negative imperative sentences
- Not only the word order of post-attributives of nouns and pronouns but also the word order within these modifiers should be adjusted
- The word order of the translations should be adjusted when some attributives are placed behind the nouns or pronouns it modifies
- Logical indirect object+ passive form of verb+ by+ logical subject

In Padideh:

- Negative sentences should be translated into affirmative ones
- Logical direct object+ passive form of verb+ to+ indirect object+ by+ logical subject

Shared in Padideh and Pars:

- The indefinite article a before a noun as a unit of measure should be translated into "یک"/yek/ (i.e., one) or "هر" /har/ (i.e., any)

3.5. Long Sentences

In Pars:

- Complex sentences with layer of subordination

Shared in Padideh and Pars:

- Simple sentences with many or long modifiers

4. CONCLUSION

The researcher arrived at the following conclusions concerning the output quality of Two English_Pesian MT programs:

A. Padideh Translator produces the best output.
B. Based on the most unsuccessfully translated testing points, the major linguistic bottlenecks of these MT programs were in the areas of Morphology, Complex Sentences, Syntactic Ambiguity & Semantic Analysis, Generation of Persian, and Long Sentences. These 5 subjects are mainly connected with the complexity in the syntactic and semantic analysis, or the difference between English and Persian.

Both programs performed fairly well in their treatment of relatively simple language phenomena, i.e., lexical coverage, phrases, simple sentences, and, to a lesser extent, morphology, but it was not the same case with their performance on the relatively complicated language phenomena, especially syntactic ambiguity and long sentences, and the generation of the target language as well. The main reason may possibly lie in the imperfection of their translation engine, which failed to take many complicated language phenomena and other difficulties in MT into consideration. In fact, the designers should have mentioned the following recommendations (or disclaimers) in the user's guides:

- Use short, declarative sentences. Declarative sentences consist of subject, verb, and object, in that order. Imperatives, wordy or convoluted sentences, and some types of questions are difficult to analyze. Sentences composed of phrases linked by conjunctions may also produce mistranslations.
- Use unambiguous words. The dictionaries allow only one translation for a word or phrase, avoid using words whose most common meaning is not the one you intended; instead, find a synonym for the specific meaning you want. For example the word *head*, when used as a noun, has several meanings. The most common meaning is the part of the body that contains the brain; director or leader is another meaning. If you mean director use the word *director* rather than *head*.
- Avoid idiomatic or informal expressions, unless you add them to the Semantic Unit Dictionaries.
- Include redundant relative pronouns, prepositions, and other words that clarify the sentence structure.

5. ACKNOWLEDGMENTS

I especially would like to offer my sincerest appreciation for great contribution of Mrs. Maria Teresa Costa Gomes Roberto, the Auxiliary Professor of the University of Aveiro, in preparing this article.

6. REFRENCES

[1] Almasoud, 'A Machine Translation (MT) or Mad Translation?' Presentation at the International Conference of Translators and Computers Session, ATA November 4-8, 1998.

[2] Craciunescu, et al. (2004). Machine Translation and Computer-Assisted Translation: a New Way of Translating? Translation Journal (TJ), Vol. 8, No. 3

[3] Examples of Nice Translation Made by Automated Translators Available on the Market' http://www.fortunecity.com/business/reception/19/mtex.htm

[4] Hutchins, John and Somers, Harold (1992). *An Introduction to Machine Translation*. London: Academic Press Limited.

[5] Hutchins, J. (1997). Evaluation of Machine Translation and Translation Tools. http://www.hutchinsweb.me.uk/HLT-1997.pdf

[6] Jurafsky, D. & Martin, J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. University of Colorado, Boulder: Prentice Hall. 2000.

[7] Ke Ping. A socio semiotic Approach to Translation. Vol. 9 N0. 3, February 2006.

[8] Kulesza, A. & Shieber, S. M. A learning Approach to Improving Sentence Level MT Evaluation, in Proceedings of the Tenth Conference on Theoretical and Methodological Issues in Machine Translation, Baltimore, 2004.

[9] Weiwei, Y. Evaluation of the Output Quality of some Prevalent English-Chinese Machine Translation Programs. http://freewebs.com/keping/P-MA-2004-YWWEVOfTheOutputQualityOfSomePrevale

A Two-stage Bootstrapping Algorithm for Relation Extraction

Ang Sun
Department of Computer Science
New York University
New York, NY, 10003, USA
asun@cs.nyu.edu

Abstract

Bootstrapping has been empirically proved to be a powerful method in learning lexico-syntactic patterns for extracting specific relations such as book-author and organization-headquarters. However, it is not clear how to adapt this method to extract more general relations such as the employment-organization (EMP-ORG) relation. Relations like EMP-ORG are actually a set of relations which involves many nominals such as executive, secretary, officer, editor and soldier. To address this challenge, we propose a two-stage bootstrapping algorithm in this paper. The first stage is a commonly used bootstrapping framework, starting with a small set of seeds (entity pairs) and a large corpus to learn relation patterns which are further used to extract more seeds. We combined it with a second stage bootstrapping which takes as input the relation patterns learned in the first stage and aims to learn relation nominals and their contexts. After the two-stage bootstrapping learning, we incorporate features extracted from learned nominals and their contexts into a state-of-the-art SVM based relation extractor and we observe a 2% gain in F-measure.

Keywords

Information Extraction; Relation Extraction; Two-stage Bootstrapping

1. Introduction

Relation Extraction is a challenging Information Extraction (IE) task which needs to find instances of predefined relations between pairs of entities. For example, there is an employment-organization (EMP-ORG) relation between entities *CEO* and *Microsoft* in the phrase *the CEO of Microsoft*. One way to combat this challenge is by applying machine learning techniques to a corpus with relation annotations. Supervised learning systems such as (Kambhatla 2004), (Zhou et al., 2005) and (Zhao and Grishman 2005) extract diverse lexical and syntactic features from an annotated corpus to train their system. While a supervised relation extraction system could achieve promising results, its portability to new domains is limited by the availability of annotated corpora. Porting such systems to new domains would involve substantial expert manual labor.

Another direction in addressing this challenging problem is using semi-supervised methods such as bootstrapping techniques. A bootstrapping-based system only needs a small set of seed examples and an unannotated corpus. These seeds are used to generate relation patterns, which in turn result in new examples being extracted from the corpus. For example, Brin (1998) uses bootstrapping for extracting pairs of book titles and authors from HTML documents. Agichtein and Gravano (2000) uses bootstrapping for extracting organization and location pairs which participate in the organization-headquarters relation from a large collection of plain texts. This paper characterizes these systems as single-stage bootstrapping since they carry out a loop from seeds to patterns and from patterns to seeds.

Previous research in using single-stage bootstrapping for relation extraction has been focusing on relations which are specific and do not seem to contain subtypes of relations. However, there are many other relations which are really a set of relations. Take EMP-ORG for example; it contains at least 3 different types of relations, executive-organization, staff-organization and other-organization (where the contexts are not sufficient enough to determine whether a person holds a managerial or general staff position in the organization). One can imagine that, compared to the organization-headquarters relation, there are more diverse ways of stating employment than headquarters of organizations. In particular, relation patterns for EMP-ORG involve more relation nominals including *executive*, *head*, *manager*, *programmer*, *editor* and many others. Suppose we start with the seed *Bill Gates* and *Microsoft*; a simple question for single-stage bootstrapping is how could we learn nominal patterns with *economist* or *editor*, involving words other than synonyms of the position of Bill Gates such as *CEO*, *chairman* or *head*?

To address this problem, we propose here a novel bootstrapping algorithm which we call two-stage bootstrapping[1]. The first stage is a commonly used single-stage bootstrapping learning framework, i.e. it starts with seeds to learn patterns and uses learned patterns to extract more seeds. The second stage bootstrapping takes as input the relation patterns learned from the first stage. It first picks out informative nominal patterns which are then used to generate queries for learning new nominals. For

[1] Two-stage bootstrapping framework is different from the multi-level bootstrapping used in (Riloff and Jones 1999) which uses a Meta-bootstrapping stage for evaluating learned seeds (NPs) and extracting the most reliable ones to restart an iteration of bootstrapping.

example, suppose we could learn a relation pattern *PERSON, former chairman of ORGANIZATION* in the first stage which is then selected by our algorithm as an informative nominal pattern; we generate a query *PERSON, former * of ORGANIZATION* to search for new nominals which could replace the wildcard. Newly learned nominals would be evaluated and high-confidence ones will be instantiated back into the patterns and passed to the first stage for extracting more seeds.

The next section first gives an overview of our two-stage bootstrapping learning framework, and then it briefly describes the *Snowball* system which serves as the basis for most of the components in our first stage bootstrapping. It also shows the details of our second stage bootstrapping, mainly explaining how to choose informative nominal patterns and how to evaluate new nominals. Section 3 will show our experiments using two-stage bootstrapping for learning relation nominals and contexts. In section 4, we extract features from learned nominals and contexts and incorporate them to improve a state-of-the-art SVM-based relation extraction system. Section 5 draws our conclusion and points out our future work.

2. A Two-stage Bootstrapping Algorithm for Relation Extraction

2.1 The Two-stage Bootstrapping Framework

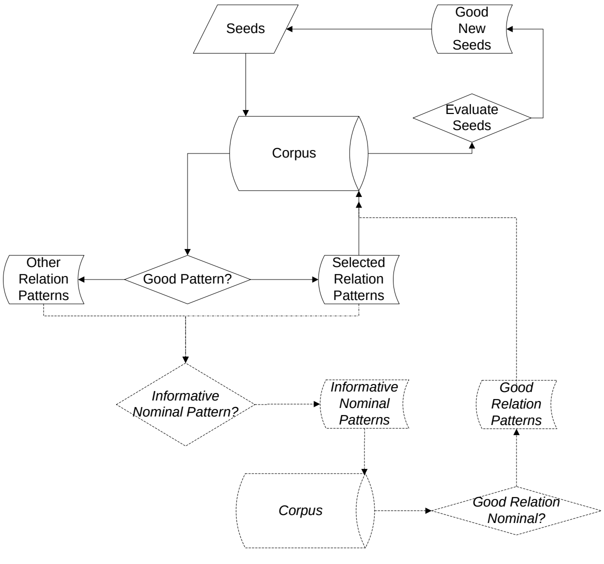


Figure 1. A two-stage bootstrapping framework

Figure 1 shows the main components of our two-stage bootstrapping learning framework. We will briefly describe the algorithm in the domain of EMP-ORG. However, it's worthy pointing out that this is a general learning framework and can be adapted to other relations. As long as a relation has a second type of evidence (relation nominals in EMP-ORG) which is associated with and could be derived and harvested from the first type of evidence in the first stage (relation patterns in EMP-ORG), the algorithm should be ready to be ported to that relation.

The two-stage bootstrapping algorithm works as follows (where the second stage is shown in *italics*):

1. Start from seeds (Person and organization pairs).

2. Search corpus for sentences containing both names.

3. Extract relation patterns from sentences.

4. Evaluate patterns:

a. Evaluate relation patterns and select high confidence ones;

b. *Select informative nominal patterns and "translate" them to relation nominal queries, e.g. PERSON, former head of ORG → PERSON, former *NN* of ORG.*

5. Search for new nominals using nominal queries.

*6. Evaluate new nominals, extract high-confidence ones and transform queries back to relation patterns, e.g. PERSON, former *NN* of ORG → PERSON, former editor of ORG* (if we learned *editor* as a high-confidence nominal).

7. Use relation patterns both from 4.a and 6 to search for new name pairs.

8. Evaluate extracted new name pairs and add the most reliable ones to the seed set.

9. If the algorithm has not reached stopping criteria, go to 2.

2.2 The First-stage Bootstrapping Framework

Our first-stage bootstrapping learning adopts most of the major components of the *Snowball* system (Agichtein and Gravano 2000).

Snowball starts with a seed set containing some initial valid seeds in the form of <*o*, *l*> such as <*Microsoft*, *Redmond*> meaning that Microsoft is an organization whose headquarters are located in Redmond. It then searches for segments of text in the text collection where *o* and *l* occur close to each other and generates patterns. A pattern in *Snowball* is a 5-tuple <*left*, *tag1*, *middle*, *tag2*, *right*>, where *tag1* and *tag2* are named-entity tags, and *left*, *middle*, and *right* are vectors associating weights with terms. For example, it generates a 5-tuple < {<*the*, *0.2*>}, *LOCATION*, {<-, *0.5*>, <*based*, *0.5*>}, *ORGANIZATION*, {} > for *the Redmond-based Microsoft*. The confidence of a pattern *P* is then estimated by the following formula. Good patterns should match more positive seeds than negative ones.

$$Conf(P) = \frac{P.positive}{(P.positive + P.negative)}$$

Top ranked patterns are then used to generate more seeds. A seed will have high confidence if it is generated by multiple high-confidence patterns. Good seeds are then used to start a new iteration of the bootstrapping learning.

$$Conf(seed) = 1 - \prod (1 - Conf(P_i))$$

We adopt Snowball's confidence measures for evaluating patterns and seeds in our first-stage bootstrapping. We represent a pattern by a 4-tuple <*order*, *tag1*, *middle*, *tag2*>, where in the domain of EMP-ORG, *order* means either PERSON-ORG or ORG-PERSON, *tag1* and *tag2* are named-entity tags, *middle* is the middle tokens between the two named-entities. For example, our system generates a 4-tuple <*PERSON-ORG*, *PERSON*, {, *former chairman of*}, *ORG*> for *Bill Gates, former chairman of Microsoft.*

2.3 The Second-stage Bootstrapping Framework

In this stage, bootstrapping first picks out nominal patterns from all the patterns returned by the first stage, then it selects informative nominal patterns for constructing relation nominal queries[2]. Queries are used to search for new nominals which will be evaluated and the top ranked ones will instantiate the queries to relation patterns. Those patterns are then used together with good patterns selected in the first stage to search for new name pairs.

2.3.1 Pick Out Nominal Patterns

We use a simple heuristic procedure to pick out nominal patterns. One thing we should mention is that we use all the relation patterns learned in the first stage as input to our procedure. The reason for including patterns which are not selected as good patterns in the first stage is that bootstrapping usually expands the pattern set in a very cautious way to guarantee learning quality. A good bootstrapping algorithm only adds the most reliable ones to grow its pattern set. However, patterns not being selected might be good nominal patterns. We will let the second stage decide the usefulness of these patterns.

The procedure first tokenizes and tags the middle part of each relation pattern with a HMM POS tagger. If a pattern contains tags *NN* or *NNS*, i.e. if it contains a common noun, then it is selected as a candidate nominal pattern. For example, P2 is a candidate while P1 is not.

P1: PERSON ,/, who/WP co/VBZ -/: founded/VBD ORG

P2: PERSON ,/, the/DT billionaire/NN chairman/NN of/IN ORG

Then for each common noun in each candidate, the following two heuristics are applied.

H1: if there is no common noun to its right, then it is the *head*. In P2, *chairman* is the *head* noun while *billionaire* is not.

H2: if H1 is true, check if the first modifier to the left of *head* is an article (a, the) or determiner (these, etc.); If not, annotate the candidate with *head* information. Articles and determiners are not good selective modifiers for learning new nominals so we do not annotate these kinds of patterns with *head*. For example, P3 would not be annotated with *head* information.

P3: PERSON ,/, the/DT chairman/NN of/IN ORG

The procedure then clusters 2 patterns together if they have the same *head* noun annotation. Finally for each cluster, if its size is larger than a threshold *t* (5 in the final experiment), then send it to the next procedure for selecting informative nominal patterns.

2.3.2 Select Informative Nominal Patterns

To estimate a pattern's selectivity, we compute **Bigram Mutual Information** (**BMI**) and **Dice** statistics between the head nominal and its direct modifier (the first modifier to its left). For example, we only consider BMI/Dice between *executive* and *director* in *PERSON, chief executive director of ORG*.

BMI, *MI(x;y)*, compares the probability of observing *x* and *y* together (the joint probability) with the probabilities of seeing *x* and *y* independently.

$$MI(x;y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

Table 1 shows an example for the bigram *senior chairman*.

Table 1. Contingency table for x=senior y=chairman

	y	-y	
x	N_{11} = 98	N_{12} = 329,653	N_{1+} = 329,751
-x	N_{21} = 337,308	N_{22} = 1,836,348,898	N_{2+} = 1,836,686,206
	N_{+1} = 337,406	N_{+2} = 1,836,678,551	N_{++} = 1,837,015,957

[2] These are queries not to an IR or Web search engine but rather to a text search engine which searches for sequences of tokens with wildcards. The queries are generated by replacing the nominal in a pattern with a wildcard and used to search for other nominals which could replace the wildcard in the pattern. For example, we generate a query *PERSON, former * of ORGANIZATION* based on pattern *PERSON, former chairman of ORGANIZATION*. We then use it to look for nominals other than *chairman* which could replace the wildcard.

We compute all these statistics in an 86-year news corpus with 1.9 billion tokens and 1,837,015,957 bigrams not crossing sentence boundaries. N_{11} is the total number of times of observing the bigram *xy*; N_{12} is the number of times *x* occurs in bigrams to the left of words other than *y*; N_{21} is the number of times *y* occurs in bigrams after words other than *x*; and N_{22} is the number of bigrams containing neither *x* nor *y*. The probabilities can be approximated by[3]: $P(x) = N_{+1}/N_{++}$, $P(y) = N_{1+}/N_{++}$, $P(x,y) = N_{11}/ N_{++}$. Then BMI can be computed as:

$$MI(x;y) = \log_2 \frac{N_{++}N_{11}}{N_{+1}N_{1+}}$$

Similarly, we can approximate *Dice(x,y)* by:

$$Dice(x,y) = \frac{2P(x,y)}{P(x)+P(y)} = \frac{2N_{11}}{N_{+1}+N_{1+}}$$

We compute BMI/Dice for each pattern in each cluster of nominal patterns and add the top ranked ones to a pattern set *S*. After the computation, we use the top ranked patterns in *S* to construct nominal queries.

2.3.3 Evaluate New Nominals

A Good nominal should be able to cross several patterns, i.e. it should match several queries. Basing a nominal's quality on one query is error-prone. So we assign a confidence score to a new nominal in the following way:

$$Conf(nom) = \sum_{i=1}^{t} \#Q$$

where *i* is the index of iterations, *t* is the maximum number of iterations during the experiment and *#Q* is the number of queries that *nom* matched during the i^{th} iteration. A nominal's confidence is updated crossing different iterations for the reason that a "loser" in the current iteration might become a "winner" in later ones. Selected new nominals are used to instantiate queries to relation patterns which will be passed to the first stage to participate in finding new name pairs.

3. Experiments for Discovering Relation Nominals

We conducted 2 experiments, one uses single-stage bootstrapping for learning relation nominals and the other one uses two-stage bootstrapping. Common parameters and tools used are summarized in Table 2.

The single-stage bootstrapping extracts all nominals from nominal patterns being picked out by the procedure described in section 2.3.1. It only learned 24 nominals, {***chairman***, *company*, ***executive***, *founder*, ***leader***, ***president***, *office*, *year*, ***director***, ***officer***, ***head***, *billionaire*, *giant*, *investor*, *group*, ***coach***, ***member***, *owner*, ***chief***, *network*, ***general***, *investment*, *opposition*, ***minister***}. One could easily judge based on common knowledge that the bold ones are correct nominals for EMP-ORG relation. All other nominals are either wrong or need context to decide.

Table 2. Common parameters and tools used

Seeds	*<Bill Gates ; Microsoft>* *<Louis Gerstner ; IBM>*
Corpus	[7]
Search engine	[7]
POS and NE tagger	Jet[4]
Maximum length of middle context	7 tokens
Maximum iteration	10

In two-stage bootstrapping, each nominal has a set of patterns matching it and each such pattern might match it several times. We assign a score to each learned nominal according to these two factors and only keep a nominal whose score is larger than 2, i.e. there are patterns/pattern matching it at least 3 times. In this way, *BMI* discovered 958 relation nominals and *Dice* 1096 nominals.

We then face the challenging problem of evaluating what we learned in the second stage. It is difficult to evaluate the results in stage two in isolation. Also, it is not feasible to directly use the ideal metric evaluation methodology suggested by *Snowball* since for one thing there is no perfect list of relation nominals available and most of the time we not only need to look at the nominal but also need to refer to the contexts to judge. Sampling evaluation normally picks the top ranked outputs or randomly picks some of the outputs to estimate the learning quality. However, we learned hundreds of nominals and thousands of contexts and we believe sampling is the not the best way to reflect the overall learning quality of our system.

We then decide to incorporate the learned lists of nominals and contexts as features into a SVM-based relation extraction system to see if its performance can be boosted or not.

[3] The reason for using N_{+1} instead of the count of x in the corpus in computing *P(x)* is that we do not count bigrams which cross sentence boundaries. Please refer to Inkpen and Hirst (2002) for a detailed description of all these statistics.

[4] Please refer to Grishman et al. (2005) and http://cs.nyu.edu/grishman/jet/license.html

4. Using Relation Nominals to Improve Supervised Relation Extraction

We first build a SVM-based relation extraction system as our baseline system trained on the annotated data from the 2004 ACE (Automatic Content Extraction) evaluation[5], which is commonly used by researchers to report and compare system performance. We then extract three types of features (to be explained in section 4.3) from nominal lists learned by two-stage bootstrapping and incorporate them into the baseline.

4.1 ACE Terminology

In ACE vocabulary, entities are objects and mentions are references to them. Entities can be of 5 types: person, organization, location, facility, and geo-political entity. Mentions have levels: name, pronoun or nominal.

The ACE Relation Detection and Characterization (RDC) task detects relations between entities. As in Example 1, there is an EMP-ORG relation between *analyst* and *the Council on Foreign Relations*, where *analyst* is a mention and referenced to the entity *Lawrence Korb*.

Example 1: *Lawrence Korb, an analyst at the Council on Foreign Relations who was assistant defense secretary under former President Ronald Reagan.*

4.2 Baseline System

Our baseline system is a duplicated system of (Zhou et al., 2005). We adopt most of their features except the *personal relative trigger word list* since it is not relevant to our EMP-ORG scenario.

Features can be characterized into 5 categories: lexical, base phrase chunking, dependency tree, parse tree and semantic resources such as country name list. Each category contains several subtypes of features. There are 41 subtypes of features in our system (the only two subtypes features we are not adapting are extracted from the *personal relative trigger word list*). Please refer to (Zhou et al., 2005) for a detailed description of features.

It's important to point out that we train our system on relation type EMP-ORG not its subtypes. We decode positive and negative instances in the following way: if 2 mentions have an EMP-ORG relation annotation in the ACE key file, we then build a positive instance; any 2 mentions within a sentence which do not have EMP-ORG relation will be built as a negative instance.

We use the official ACE 2004 training and evaluation data from LDC for experiment. We exclude the 8 fisher transcript files from training data. So we use in total 635 files and there are 2,874/773,412 positive/negative instances.

We use the SVM-light[6] package as our machine learning method. We use a linear kernel and do 10-fold cross-validation in our baseline and all the other experiments.

4.3 Features Extracted from Nominal Lists

We extract three types of features from our two-stage bootstrapping learned nominal lists.

4.3.1 InNomList

This is a binary feature which checks whether the *head* of *mention 1* is in our learned list or not. We combine it with the entity types of both mentions to prevent the feature from being too general. In Example 1, the *head* of *mention 1* is *analyst* and it is in our learned nominal list. So for Example 1, we construct the following feature:

InNomList=PERSON-ORG-true

4.3.2 ContextNomList

For each learned nominal in our lists, we assume it is the *head* of *mention 1* in a learned pattern. We then generate a list *L* of words between this nominal and the *head* of *mention 2* which is an ORG in our pattern. For example, *L = {at, of, for}* for nominal *analyst* and its associated patterns { *PERSON, an *NN* at ORG; PERSON , chief *NN* of ORG*; *PERSON , managing *NN* at ORG*; *PERSON , chief financial *NN* for ORG* }

Given a positive/negative instance, we first extract the words between the heads of both mentions; Then we check if the words are in *L* or not. In Example 1, *at* is the word between *analyst* and *the Council* and is in *L*, so we construct the following feature:

ContextNomList=PERSON--ORG--at (combined with entity types)

4.3.3 ModNomList

For each learned nominal, we generate a list *L* of its modifiers from its associated patterns. *L = { an, chief, managing, financial }* for the *analyst* example. Given an instance, we extract the words before the head of *mention 1* and check if they are in our modifier list or not. We generate the following feature for Example 1:

ModNomList=an-PERSON (combined with entity type of mention 1)

4.4 Experiments and Results

We conducted 4 experiments. In the rest of this paper, we will refer to the system which added to *Baseline* features from ***BMI*** learned nominal list as *System BMI*, the system which added features from ***Dice*** learned list as *System Dice* and the system which added features from the merged list of ***BMI*** and ***Dice*** as *System Combined*.

[5] http://www.itl.nist.gov/iad/mig//tests/ace/

[6] http://svmlight.joachims.org/

Figure 2, 3 and 4 show Precision, Recall and F-measure of our experiments, where 1/2/3/4 mean *System Baseline/BMI/Dice/Combined*.

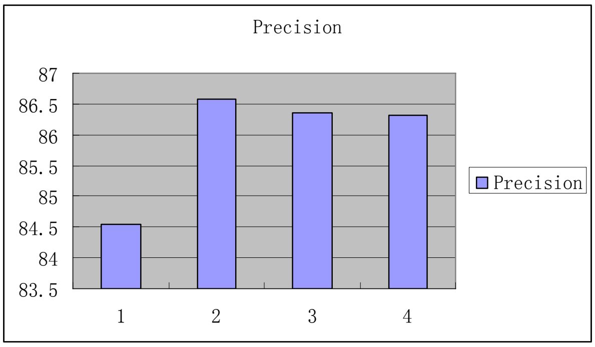


Figure 2. Precison

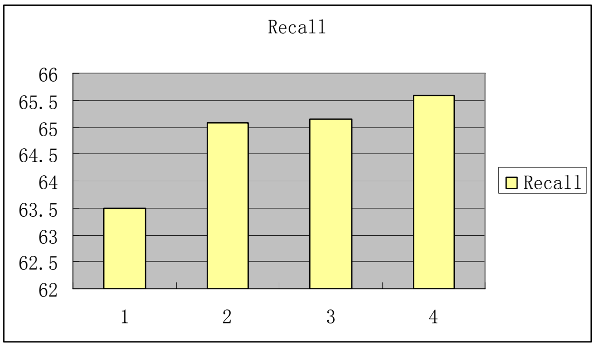


Figure 3. Recall

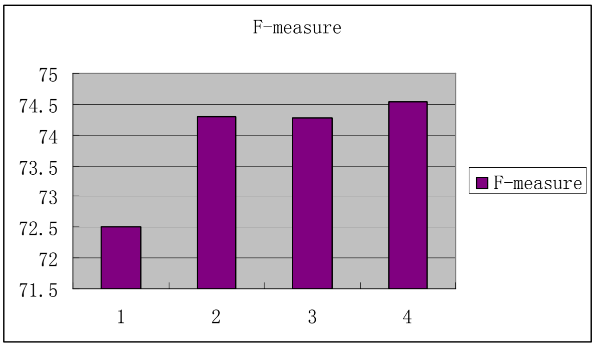


Figure 4. F-measure

We first notice that our baseline achieves similar results as reported in (Yong and Su 2008) whose baseline is also a duplicated system of (Zhou et al., 2005)[7]. Though, we should point out that we achieve a slightly higher precision and a lower recall than (Yong and Su 2008). This is first because we use different experiment settings. Also, it is probably caused by different tools used for generating features. We use the same Perl script[8] used in (Zhou et al., 2005) to derive base phrase chunk information from full parse tree. But we use Jet for tokenization and the Charniak parser for full parsing.

The results show that:

- Both *MI* and *Dice* improves the Baseline while *MI* achieves slightly higher precision than *Dice* and *Dice* achieves slightly better recall than *MI*. When we check our nominal list, we found that *MI* is more cautious than *Dice* in expanding patterns for a nominal. Table 3 shows the top 10 ranked nominals and the number of associated patterns for both *MI* and *Dice*. It may be that *Dice* achieves better recall in part because there are more patterns being used for generating features. However, *Dice* would sacrifice precision a little bit because it might also include some bad patterns and thus some bad nominals which are generated by these bad patterns.
- The merged list gives the best recall and F-measure. There are 200 nominals from the merged list which are used during feature decoding while there are 152/156 nominals from *MI/Dice* being used. We can also imagine that the merged list uses more patterns for feature decoding.

Table 3. Top 10 ranked nominals and number of associated patterns

Nominal (*MI*)	Number of patterns (*MI*)	Nominal (*Dice*)	Number of patterns (*Dice*)
executive	109	executive	126
scientist	32	economist	66
economist	28	analyst	58
editor	24	editor	46
architect	24	officer	43
minister	23	scientist	40
justice	20	engineer	37
counsel	20	architect	36
judge	20	investigator	35
designer	20	counsel	34
Total:	320		521

[7] Zhou et al. (2005) reported result for ACE 2003 data, not ACE 2004 data.

[8] http://ilk.uvt.nl/team/sabine/chunklink/README.html

- Although there are some differences between the MI list and Dice list, they give similar improvements and the combined list gives even more improvements. This suggests that MI and Dice lists are complementary to each other and both of them are reliable.

5. Conclusions and Future Work

Using bootstrapping to extract relations which are normally general and contain subtypes of relations is challenging. This paper proposes a two-stage bootstrapping learning algorithm for addressing this problem. We show a case study in EMP-ORG and observe 2% F-measure improvement when we incorporate features extracted from two-stage bootstrapping learned nominals and context into a supervised relation extraction system.

Our immediate future work involves testing this method on more relation types. Our current system relies on tokens between name pairs. Incorporating parsing information into two-stage bootstrapping would be challenging yet interesting future work.

6. Acknowledgements

This research was supported by the National Science Foundation under Grant IIS-00325657. This paper does not necessarily reflect the position of the U.S. Government. We would like to thank Prof. Ralph Grishman for his research guidance. We would also like to thank Prof. Satoshi Sekine, Prof. Adam Meyers and Prof. Heng Ji for their useful suggestions.

7. References

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain text collections. In Proceedings of the 5th ACM International Conference on Digital Libraries, 2000.

[2] S. Brin. Extracting patterns and relations from the World-Wide Web. In Proceedings of the 1998 International Workshop on the Web and Databases (WebDB'98), March 1998.

[3] R. Grishman, D. Westbrook and A. Meyers. 2005. NYU's English ACE 2005 System Description. ACE 2005 PI Workshop. Washington, US.

[4] DZ. Inkpen and G. Hirst. 2002. Acquiring collocations for lexical choice between near synonyms. In Unsupervised Lexical Acquisition: Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX), pp. 67–76, Philadelphia, Pennsylvania.

[5] N. Kambhatla. 2004. Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics.

[6] E. Riloff and R. Jones. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99).

[7] S. Sekine. 2008. A Linguistic Knowledge Discovery Tool: Very Large Ngram Database Search with Arbitrary Wildcards. In proceedings of the 22nd International Conference on Computational Linguistics, Manchester, England.

[8] SWK Yong and J. Su. An Effective Method of Using Web Based Information for Relation Extraction. In proceedings of 3rd International Joint Conference of Natural Language Processing (IJCNLP2008), P350-357, Hyderabad, India.

[9] S. Zhao and R. Grishman. 2005. Extracting relations with integrated information using kernel methods. In Proceedings of ACL.

[10] G. Zhou, J. Su, J. Zhang and M. Zhang. 2005. Exploring Various Knowledge in Relation Extraction. In proceedings of 43th Annual Meeting of the Association for Computational Linguistics. USA.

Context Driven XML Retrieval

Aneliya Tincheva
IPP-BAS
25A Acad. G. Bonchev Str.
Sofia 1113
nelitincheva@gmail.com

Abstract

This paper presents a data-centric approach to XML information retrieval which benefits from XML document structure and adapts traditional text-centric information retrieval techniques to deal with text content inside XML. We implement our ideas in a configurable, general purpose XML retrieval library which can be tuned to operate on multilingual XML resources with different structure and can be used to extract relevant document fragments with different granularity according to user preferences. We present a rich query format and an algorithm for indexing and query processing.

Keywords

XML Retrieval, IR, XML-IR, XPath, document fragment, indexing schema, full-text indexing

1. Introduction

The popularity of the e**X**tensible **M**arkup **L**anguage (XML) has led large quantities of structured information to be stored in this format. Due to this ubiquity, there has lately been interest in information retrieval (IR) from XML. XML-IR presents different challenges than retrieval in text documents due to the semi-structured nature of the data. The goal is to take advantage of the structure of explicitly marked up documents to provide more focused retrieval results. For example, the correct result for a search query might not be a whole document, but a document fragment. Alternatively, the user could directly specify conditions to limit the scope of search to specific XML nodes. Previous work [2, 4] addresses several challenges specific to retrieval from XML documents:

(1) *Granularity of indexing units* (Which parts of an XML document should we index?)

(2) *Granularity of the retrieved results*(Which XML nodes are most relevant?)

(3) *Ranking of XML sub-trees* (How should the ranking depend on the type of enclosing XML element and term frequency/inverse document frequency (*tf-idf*)?)

The aim of this work is to define an approach for XML retrieval that can be used for indexing and search independently of the document structure. We call our approach *context driven XML retrieval* because indexing and search operate on parts of XML documents called *contexts.* These contexts represent searchable and retrievable parts of an XML document, and for us the IR problem can be viewed as the extraction of contexts that match some search criteria. Traditional IR is a special case of XML-IR where the context has to be a whole document. Narrower contexts could be separate XML elements or their combinations. Our setting assumes knowledge of the XML document structure and the retrieval requirements. Thus an administrator creates indexing and retrieval rules for different XML document corpora. Each corpus requires different indexing rules to define contexts and relations between them – document fragments referable at search time. Using this context driven approach we address challenges (1) and (2). Concerning ranking (3), we employ a strategy which combines the unstructured and structured IR scoring techniques. In the paper we present a scalable index structure, indexing, search algorithms, indexing rules and query language format. We implement our ideas in a general purpose XML retrieval library that can be integrated in different kinds of applications: web applications, standalone systems, web services.

The rest of the paper is organized as follows: Section 2 describes related work; Section 3 describes motivation; Section 4 presents implementation details. Section 5 concludes the paper and describes future work.

2. Related work

XML retrieval systems vary according to the query language, index structure, document preprocessing, indexing and scoring algorithms they employ. A great variety of XML query languages already exist. Standard ones proposed by W3C are XPath and XQuery. Unfortunately, they do not reflect IR properties such as weighing, relevance-oriented search, data types and vague predicates, structural relativism [1, 14]. Amer-Yahia et al. [4] classifies XML query languages into three classes: keyword query languages (KQL) [5, 6, 12, 13]; tag & KQL [6]; path & KQL [7, 8, 12]; XQuery & KQL [10]. The query language we introduce is a *path and KQL* in XML format, and most related to XPath 2.0, XIRQL, XXL, NEXI CAS queries. Different term and structure statistics are implemented in separate XML-IR systems. We share the idea of Mass and Mandelbrod [11] that an XML index consists of a set of separate full-text indices. For full-text search we use the Apache search API Lucene [9].

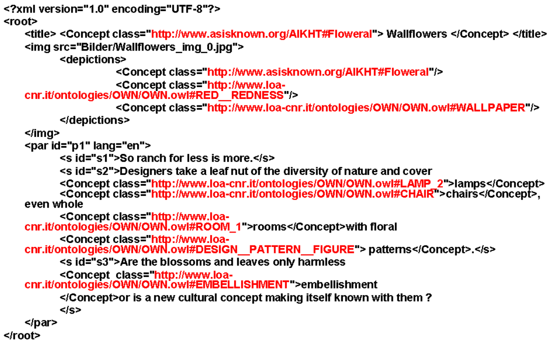
The context driven approach we present can be classified as **C**ontent-**A**nd-**S**tructure (CAS) retrieval under the system developed by the Initiative for the Evaluation of XML Retrieval (INEX) [12].

3. Motivation

In addition to the growing interest in XML retrieval, we had a practical need for an IR system for XML documents. In order to aid annotation efforts, we needed a platform independent search engine that could be tuned for specific applications. Since the document structure was known and important, we wanted to create indexing and retrieval rules to improve retrieval. The system we present allows exactly such application-specific indexing and search.

4. Context Driven XML Retrieval

An *XML document* is a tree-like data structure which consists of three node types: *elements*, *attributes*, *character data/text*. XML document tree nodes are instances of *elements* called *tags/markups*, which can be either empty or have nested *elements* or *text nodes*. *Attributes* are name-value pairs attached to *tags*. Figure 1 is an example of a textile multimedia XML document created by our group for the purposes of the AsIsKnown project [15]. The example document contains text and images annotated with *concepts* from a textile knowledge base. Text is delimited in sentences which are organized in paragraphs.



```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <title> <Concept class="http://www.asisknown.org/AIKHT#Floweral"> Wallflowers </Concept> </title>
    <img src="Bilder/Wallflowers_img_0.jpg">
        <depictions>
                <Concept class="http://www.asisknown.org/AIKHT#Floweral"/>
                <Concept class="http://www.loa-
    cnr.it/ontologies/OWN/OWN.owl#RED__REDNESS"/>
                <Concept class="http://www.loa-cnr.it/ontologies/OWN/OWN.owl#WALLPAPER"/>
        </depictions>
    </img>
    <par id="p1" lang="en">
        <s id="s1">So ranch for less is more.</s>
        <s id="s2">Designers take a leaf nut of the diversity of nature and cover
        <Concept class="http://www.loa-cnr.it/ontologies/OWN/OWN.owl#LAMP_2">lamps</Concept>
        <Concept class="http://www.loa-cnr.it/ontologies/OWN/OWN.owl#CHAIR">chairs</Concept>,
    even whole
        <Concept class="http://www.loa-
    cnr.it/ontologies/OWN/OWN.owl#ROOM_1">rooms</Concept>with floral
        <Concept class="http://www.loa-
    cnr.it/ontologies/OWN/OWN.owl#DESIGN__PATTERN__FIGURE"> patterns</Concept>.</s>
        <s id="s3">Are the blossoms and leaves only harmless
        <Concept  class="http://www.loa-
    cnr.it/ontologies/OWN/OWN.owl#EMBELLISHMENT">embellishment
        </Concept>or is a new cultural concept making itself known with them ?
        </s>
    </par>
</root>
```

Figure 1. Multimedia XML Document

There exist W3C standard languages for navigation through XML documents (XPath) and querying (XQuery). We employ XPath in the implementation of our framework. After the evaluation of an Xpath expression, a set of XML nodes are retrieved. For example, if we want to extract the sentences which contain the word *pattern* from our example XML document in Figure 1, we can use the XPath expression: **//s[contains (descendant::text(), "pattern")]** . If we need sentences containing the document's title, we need variables to store temporary results and be used in the search expression. We deal with the problem by using an extension of XPath with variables. The expression below extracts the sentences containing the text of the title:

{x:=/title/descendant::text()}/s[contains(descendant::text(), $x)]

We set the variable *x* to be equal to the document title text. The XPath engine extracts the sentences whose text contains the value of the *x* variable.

4.1 System architecture

We implement our system in an XML retrieval library. Each document corpus has a separate XML index in a central *XML Index Repository*. Each index has its own *model*, defined in an *indexing schema.* The indexing schemas specify how to extract indexing units (XML subtrees) called *contexts* from XML documents with a common structure and defines how the separate *contexts* are related. Our basic assumption is that a *context* is a document fragment whose content is indexed in several text fields. A *field* is a name-value pair whose value is character data. *Contexts* and *fields* can be referred to in search queries. An indexing schema is added to an empty index on its creation. Existent XML indices are populated with documents according to the definitions in their corresponding indexing schema. For each context defined in the indexing schema we create and populate a full-text index called *context index.* The rest of the subsection gives an overview on the system architecture (Figure 2)

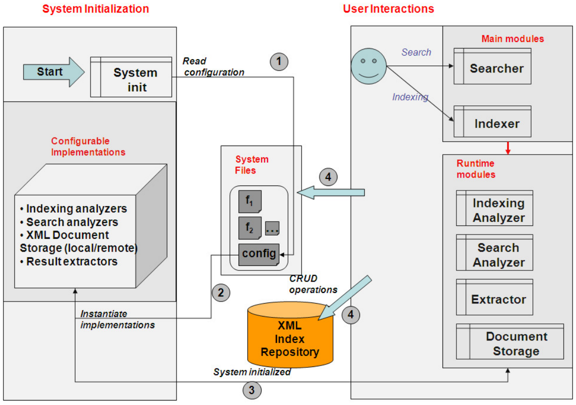


Figure 2. System Architecture

The search engine lifecycle has two sequential phases: *system initialization* and *user interactions*. When the system is started a processing module reads system configurations from a configuration file. This file specifies an indexing/search analyzer, document storage, and result extractor implementation classes. The opportunity to configure different implementations makes our framework highly adaptable to various search scenarios. We can use this to tune text analysis, document access policy, result formatting and extraction. Instances of the configured implementations are created by reflection and are made available at runtime. The indexing and retrieval tasks are performed by the *indexer* and *searcher* operational modules which manipulate the XML indices, system files and interact with the already instantiated objects.

4.2 Indexing schema and index structure

Each XML index consists of an *indexing schema* and one or more full-text indices. The *indexing schema* is an XML document which defines indexing and extraction rules. Central concepts are *context* and *field*, as defined in the previous subsection. Their usage is clarified with the following example. Assume that we have a corpus with documents with the same structure as the one in Figure 1 and we want to retrieve particular paragraphs/ images. For the purpose we define 2 independent contexts: *paragraph* and *image*. Our aim is to search for a combination of text matches and concepts. We need to create two fields for the *paragraph* context (*text* and *concept*) and one for the *image* context (*concept*). We add one more requirement - we want to extract paragraphs whose adjacent paragraphs and images comply with supplementary search criteria. In this case we need to define some kind of relation between a paragraph and its adjacent paragraphs and images. We call this type of relation *coordination between contexts*. In the indexing schema the contexts are defined as *context* elements, *fields* are elements nested in *context* elements and *coordination between contexts* is expressed by nesting *context* elements.

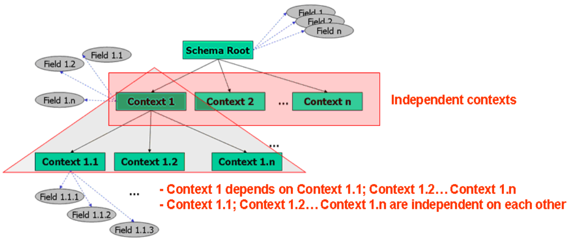


Figure 4. Indexing Schema Logical Tree View.

Each *context* and *field* element has an identifier and is associated with an XPath expression. Indexing schema elements are relative to their parent elements, i.e. their identifiers are unique within the scope of their parent element and their XPath expressions are applied relative to the nodes extracted by the XPath expressions of their parent element. The *schema root* element denotes the *default context*, i.e. the whole XML document. *Field* elements have no nested elements and can be boosted. For *context* elements with child *field* elements we create separate full-text indices in the XML index repository. Figure 5 illustrates the Lucene index (full-text index) and the XML index structures.

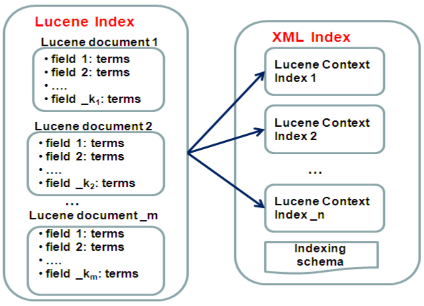
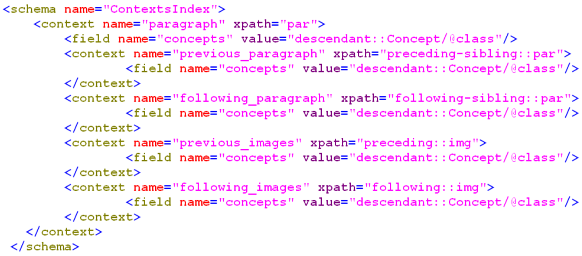


Figure 5. Lucene and XML Index Structure

The Lucene index (left) is an inverted index consisting of a set of Lucene documents. Each Lucene document carries a unique identifier and contains fields. The XML index (on the right) contains an indexing schema document and a set of Lucene context indices. They are populated with Lucene documents with identifiers which encode the system identifier of the XML document, path to an XML context node and paths to its related XML nodes. The Lucene documents are populated with fields as defined in the indexing schema document. An illustrative example is to index the document in Figure 1. We want to search by concept to extract paragraphs that match one concept pattern with adjacent paragraphs or images matching another pattern. Such a relation between structural document units allows us to create complex search queries. An indexing schema satisfying our requirements is given in Figure 6.



```
<schema name="ContextsIndex">
    <context name="paragraph" xpath="par">
        <field name="concepts" value="descendant::Concept/@class"/>
        <context name="previous_paragraph" xpath="preceding-sibling::par">
                <field name="concepts" value="descendant::Concept/@class"/>
        </context>
        <context name="following_paragraph" xpath="following-sibling::par">
                <field name="concepts" value="descendant::Concept/@class"/>
        </context>
        <context name="previous_images" xpath="preceding::img">
                <field name="concepts" value="descendant::Concept/@class"/>
        </context>
        <context name="following_images" xpath="following::img">
                <field name="concepts" value="descendant::Concept/@class"/>
        </context>
   </context>
 </schema>
```

Figure 6. Example indexing schema

The index created according to the definitions of the indexing schema on Figure 6 contains five full-text indexes (one for each context with a field). If the document in Figure 1 has a system identifier *f1*, the content of the separate full-text indexes after the indexing would be:

Lucene context index	Lucene document IDs	Field content
paragraph	f1_p#1	The concept URIs in the first paragraph.
paragraph→*previous_paragraph;* (no data added)*		
paragraph→*following_paragraph* (no data added)*		
paragraph →*previous_images*	f1_p#1@@@f1_img#1	The concept URIs in the first image.
paragraph →*following_images* (no data added)*		

Table 1. Content of Lucene indices for the example XML Index[1]

Indexing is incremental. When a new XML documents is added to an XML index, its Lucene context indices are updated by creating and populating Lucene documents.

4.3 Indexing algorithm

Below (Figure 7) is listed the indexing algorithm which is recursive in nature. The recursive structure is inherited from the nesting of contexts.

```
                              Index XML document
1. Index default context fields → call F1
2. Index nested contexts → call F2
                              F1: Index context fields
1.  Create an empty Lucene document
2.  Foreach field ← Fields
      2.1 extract field's content from the XML document and create a Lucene field
      2.2 add the Lucene field to the Lucene document
3.  Add the Lucene document to the index
                              F2: Index nested contexts
foreach context ← Contexts
{
    1. Extract the context XML nodes from the XML document → contextXMLNodes
    2. foreach xmlNode: contextXMLNodes
    {
          5.1 Get the absolute path to the xmlNode → absolutePath
          5.2 Get the indexing schema fields for the context → fields
          5.3 Encode the Lucene context document ID → documentID
          5.4 Create a context index and  call F1 for the context fields;
          5.5 Get the nested  indexing schema contexts for the context → nestedContexts
          5.6 if (nestedContexts is NOT empty)
                    call F2 for the nestedContexts
    }
}
```

Figure 7. Search procedure

4.4 Query syntax and search algorithm

Search is performed within a single index in order to retrieve relevant content. The search criteria are specified in a query XML document whose format is presented below. *Contexts* and *fields*[2] are referred to in search queries. Figure 8 (below) illustrates the query structure.

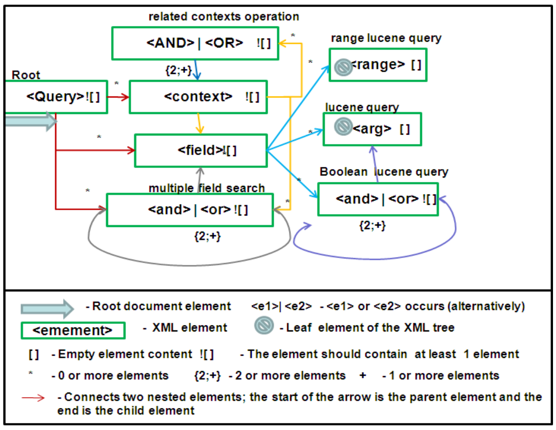

related contexts operation
<AND> | <OR> ![]
{2;+}
Root
<Query>![]
<context> ![]
<field>![]
multiple field search
<and> | <or> ![]
{2;+}
range lucene query
<range> []
lucene query
<arg> []
Boolean lucene query
<and> | <or> ![]
{2;+}
*
- Root document element
<e1>| <e2> - <e1> or <e2> occurs (alternatively)
<ement> - XML element
- Leaf element of the XML tree
[] - Empty element content ![] - The element should contain at least 1 element
* - 0 or more elements {2;+} - 2 or more elements + - 1 or more elements
- Connects two nested elements; the start of the arrow is the parent element and the end is the child element

Figure 8. Search query structure

Queries have a recursive structure similar to the indexing schema. The *context* and *field* elements in queries refer to corresponding elements in the indexing schema by ID. The parent-child element relationship for both contexts and fields **should** follow the order in the indexing schema. The element content of query *context* elements consists of:

- *field* element(s) – their element content is transformed to a Lucene query. The supported full-text queries for a field include term, phrase, fuzzy, Boolean, span, wildcard, range queries. Search results are sorted by *tf-idf*.
- *AND; OR* | *and;or* elements – recursive multi-argument operations denoting *intersection* and *union* of search results returned for sets of *context* arguments (*AND; OR*) or *field* arguments (*and; or*). In either case, results are sorted by *tf-idf.*

The search algorithm is presented below (Figure 9).

```
                              XML Index search
Iterate query root children
    1. For all children = <context> → call F1 : result map with key-value pairs <context ID, result list>
    2. If child= <and> | <or> | <field> → call F2 : result list
Return → result map + <global context, result list> entry
                              F1:Context search
Foreach context element:
    Iterate context children
          1. If  child = <AND> | <OR>  → call F3: context result list
          2. If child = <and> | <or> | <field> → call F2 → result list
          3. result = result + context result list + result list (sort by tf-idf)
Return → result
                              F2: Field Search
If current element = <and> | <or> → push operator in stack
    1. Iterate current element children
          1.1. For all <field>  elements → lucene search : push result list in stack
          1.2. For all  <and> | <or> elements → push operator in stack → call F2
    2. Process stack → pop the peak: if peak = result list → add it to temp list, else process operation <and> |
    <or> on temp list (sort by tf-idf) and push back the result {do while the stack  is not empty}
    3. Return → temp
Else process lucene search → Return result
                              F3: Context operations search
If current element = <AND> | <OR> → push operator in stack and iterate current element children
          1.1. For all  <context> elements from children → call F1 : push result list in stack
          1.2. For all  <AND> | <OR> elements from children → push operator in stack → call F3
    2. Process stack → pop the peak: if peak = result list → add it to temp list, else process operation <AND> |
    <OR> on temp list (sort by tf-idf) and push back the result {do while the stack  is not empty} →
    3. Return → temp
Else call F1 → Return result
// !NOTE: the leaf context elements should have only <and> | <or> | <field> child elements, else endless cycle
```

Figure 9. Search procedure

Our search algorithm performs depth first search in contexts. If a context has a descendant context then a recursive call is made. The bottom of the recursion is reached when no more context descendants are available. For each separate context having *fields* we perform search in its corresponding Lucene index.

4.5 Experiment

We evaluated the search engine on a corpus of 135 multimedia documents in XML format, in 3 separate indexes. Our goal was to evaluate indexing and search performance and retrieval relevance. The documents are multilingual (English and Italian) and contain markers for paragraphs, images, and concept annotation for both images and text. The total number of text terms in the corpus is 117 307 and the total number of concept annotations is 9547. The paragraph text is indexed in index (I_1). Concept annotations in paragraphs as well as

[1] The @@@ is a separator between identifiers of dependent contexts. The Lucene document identifier f1_p#1@@@f1_img#1 (row 4) encodes that the first paragraph is related to the first image with a *previous_images* relation.

[2] Contexts and fields as defined in the indexing schema for the index in which we are searching.

the concepts in the preceding and following paragraphs and images are indexed in index (I_2). Finally, the text of paragraphs, besides the concepts, is indexed in index (I_3). For text analysis we used the Lucene *StandardAnalyzer*. This class uses a Java CC-based grammar to tokenize alpha-numeric strings, acronyms, company names, e-mail addresses, computer host names, numbers, words with an interior apostrophe, serial numbers, IP addresses, and CJK (Chinese Japanese Korean) characters. As we see in Chart 1, relatively little time is consumed in the creation of I_1. Increasing the schema complexity degrades indexing performance.

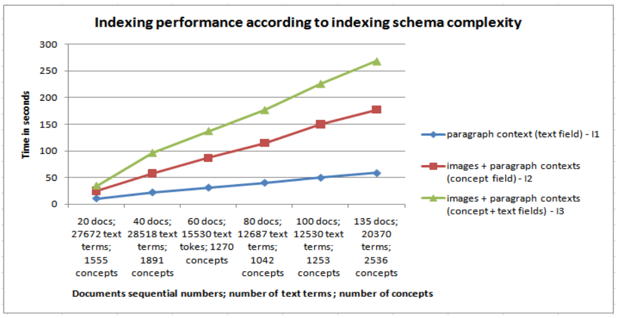


Chart 1. Indexing performance

This degradation was not a major concern. We focus on estimating:

(1) Does indexing performance degrade *drastically* for complex indexing schemas?

(2) To what extent does the search performance decreases for deeper and broader indexing schemas?

(3) What is the benefit in precision and recall from querying indexes with more complex schemas?

Since indexing can be performed in the background and offline, we are only interested in drastic degradation for (1). Schema complexity appears not to affect search performance significantly. We see in Chart 1 that schema complexity only moderately affects indexing performance. With the most complex schema all documents are indexed in under 5 minutes. The estimation issues (2) and (3) are the important ones to be evaluated for a search application. Precision and recall mostly depend on the preciseness and quality of the query, availability of metadata in XML documents, the metadata indexing and querying strategy. The results were more than satisfactory addressing issue (2). For all queries for each index, we obtained results in under 0.5 seconds. We did not evaluate the system on a standard dataset. On the basis of the experiment with $I_{1,}$ I_2 and I_3 we concluded that precision and recall increase for more complex schemas (3). The precision when querying I_2 is higher than the one for I_1, because it retrieves conceptually relevant documents either in English or Italian. Although many of the text terms are not annotated with concepts and the variety of queries is limited, the recall for I_2 is comparable with the one from I_1. The retrieval from I_3 is with the highest precision and recall, since it combines advantages of I_1 and $I_{2.}$

5. Conclusion

The aim of this work was to define a user guided approach to XML retrieval that could be used for indexing and search in XML document corpora independent of document structure. We implemented our ideas in a platform independent search engine framework that combines structured and unstructured retrieval techniques and can be integrated in different kind of applications. We ended up with a middle layer component which so far is integrated into prototype systems created for the LT4eL [16] and AsIsKnown [15] research projects. Future work includes running experiments with the test collections created for the INEX competition. Our future goals include integration of automatic language detection and format conversion to XML. We also intend to implement and integrate different tokenizers and analyzers. A future aim and big challenge for us is to adapt and integrate the framework in a web search engine.

Acknowledgements

This work was supported by the Institute for Parallel Processing (Bulgarian Academy of Science) and granted by the European Projects AsIsKnown (FP6-028044) and LTfLL (FP7-212578). I thank to my advisor Kiril Simov who laid the foundation of the work and assisted me during the past year and a half in the process of design, analysis, development and integration of the XML-IR framework.

6. References

[1] N. Fuhr: XML Information Retrieval and Information Extraction, University of Dortmund, Germany, 2003

[2] Ch. D. Manning, P. Raghavan, H. Schütze: Introduction to Information Retrieval, ISBN: 0521865719,Cambridge University Press. 2008.

[3] D. Carmel, N. Efrati, G. M. Landau, Y. S. Maarek, and Y. Mass. An Extension of the Vector Space Model for Querying XML Documents via XML Fragments. Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval, 15. August 2002

[4] S. Amer-Yahia, M. Lalmas: XML search: languages, INEX and scoring. SIGMOD Record 35(4): 16-23 (2006)

[5] L. Guo, F. Shao, C. Botev, J. Shanmugasundaram.XRANK: Ranked Keyword Search over XML Documents.SIGMOD 2003.

[6] S. Cohen, J. Mamou. Y. Kanza, Y. Sagiv. XSEarch: A Semantic Search Engine for XML. VLDB 2003

[7] A. Theobald, G. Weikum. The Index-Based XXL Search Engine for Querying XML Data with Relevance Ranking. EDBT 2002

[8] A. Trotman, M. Lalmas. The Interpretation of CAS. INEX 2005

[9] E. Hatcher, O. Gospodnetić, and M. McCandless, “Lucene In Action”, ISBN: 1932394281, May 2008

[10] XQueryFull-Text: http://www.w3.org/TR/xpath-full-text-10/

[11] Y. Mass, M. Mandelbrod. Retrieving the most relevant XML Components. INEX 2004

[12] INEX 2009: http://www.inex.otago.ac.nz/

[13] N. Roussopoulos, S. Kelley, F. Vincent, Nearest Neighbor Queries, 1995

[14] N. Fuhr and G. Weikum. Classification and Intelligent Search on Information in XML. Bulletin of the IEEE Technical Committee on Data Engineering, 25(1), 2002.

[15] AsIsKnown: http://www.asisknown.org/

[16] LT4eL: http://www.lt4el.eu/

Hierarchical Discourse Parsing based on Similarity Metrics

Ravikiran Vadlapudi, Poornima Malepati, Suman Yelati
International Institute of Information Technology Hyderabad,
Gachibowli Hyderabad India 500019
{*ravikiranv, mpoornima, suman.yelatipg08*}@*research.iiit.ac.in*

Abstract

Attentional State Theory and Rhetorical Structure Theory are two predominant theories of discourse parsing. Combining these two approaches, in this paper, we describe a novel approach for discourse parsing. The resulting discourse tree structure retains following properties: structure of purpose from Attentional State Theory and relations between sentences from Rhetorical Structure Theory. We demonstrate the utility of our model by constructing a summarization system.

Keywords

Discourse Parsing, Attentional state theory, Rhetorical structure theory, Coreference, Cohesion, Similarity metrics, Sentence similarity

1 Introduction

Discourse parsing is crucial for parsing texts in Natural Language where each sentence has a purpose modelled by relationship with other sentences in the discourse. The main objective of discourse parsing is to generate a valid discourse structure which captures this purpose. Discourse parsing finds its applications in a variety of fields some of which include summarization (focused summaries), dialog systems (language generation) and information retrieval (Question Answering (QA) systems).

A few theories have been proposed for structuring a discourse, two of which are Attentional State Theory (AST) [7] and Rhetorical Structure Theory (RST) [12]. Attentional State Theory (AST) stresses the role of purpose in processing the discourse. It contains three components, namely, a linguistic structure, an intentional structure and an attentional state. The linguistic structure models the structure of a sequence of sentences in a discourse. A sequence of sentences in a discourse are aggregated into discourse segments. Each sentence serves a purpose in a discourse segment and in turn each discourse segment serves a purpose (Discourse Segment Purpose, DSP) with respect to the overall discourse. This structure of purpose of a discourse segment (and in turn the sentence) is modelled by intentional structure. To capture the structure of purpose, discourse segments are related using two relations namely, *dominance* and *satisfaction-precedence.* If the purpose of a discourse segment (DSP1) contributes to the purpose of another discourse segment (DSP2), then, DSP2 *dominates* DSP1. If the purpose of a discourse segment (DSP1) has to be satisfied before the purpose of another discourse segment (DSP2), then, DSP2 *satisfaction-precedes* DSP1. At any given point of time, the discourse segment purpose under focus is kept track of by attentional state.

Rhetorical Structure Theory (RST) represents the structure of a discourse as a hierarchical tree diagram based on the relationship between sentences/text spans (nodes). The relation between these nodes can be of two types: *symmetric* and *asymmetric*. A *symmetric* relation relates two or more nodes labelled nuclei, each of which are equally important in realizing the writer's communicative goals. An *asymmetric* relation relates two nodes, a nucleus and a satellite, the nucleus being more important of the two and the satellite modifying the nucleus based on the particular relation. Due to the hierarchical structure of RST, summarization is fairly simple and efficient with respect to quality of output: the summary generated by omitting satellites of the tree after a certain depth would outline the main points of the text.

Few earlier systems aimed at discourse parsing are: A rule based system RASTA (Rhetorical Structure Theory Analyser) [3] is a component of Microsoft English Grammar that constructs RST analysis of texts. For each relation type, they define a set of rules between nodes, satisfying which, the nodes are related by that relation type. Another rule based system [15], develops a Unified Linguistic Discourse Model (U-LDM) for parsing a discourse. They perform discourse segmentation using discourse semantics and build a discourse tree based on syntactic, semantic and lexical rules. One of the machine learning approaches for discourse parsing [16] learns on discourse annotated corpora. They generate a rhetorical structure tree of a sentence based on the learned models of RST-DT 2002 corpus. Work done in [6] extend LTAG approach to discourse parsing, the D-LTAG. An incremental discourse parsing model has been proposed in [4] which aims at building a rhetorical structure of discourse using veins theory [5]. In [9], an automatic generation of discourse tree has been proposed. The sentences are nodes of the discourse tree and the edges between the sentences define a relation. The relationship type is defined using clue expressions. As per our knowledge, there are no freely available annotated discourse tree banks which makes machine learning approaches in-feasible. Rule based systems require tedious analysis of discourse and the rules might not be generic which restricts the domain of usage of the system. Therefore it is necessary to develop an approach which is generic and is un-supervised.

In this paper we combine the ideas of Attentional State Theory and Rhetorical Structure Theory and propose a novel scheme for discourse parsing. We incorporate the features of RST into AST so that we can understand the role of a sentence better and in turn create a better interpretation of the discourse which increases the quality of applications built on it. The discourse tree we build retains following properties: structure of sequence of sentences (linguistic structure) and structure of purpose (intentional structure) from Attentional State Theory and relations between sentences/text spans from Rhetorical Structure Theory. We develop a discourse tree structure based on similarity measures between sentences, identify discourse segments from the discourse tree structure and define relationships between discourse segments and between sentences of discourse segments. We demonstrate the utility of our model by constructing a summarization system. The paper is structured as follows: First we describe our system of discourse parsing in detail in section 2. In section 3 we develop a simple method of summarization based on our system with results. In section 4, we talk about future work and conclude.

2 Our Model

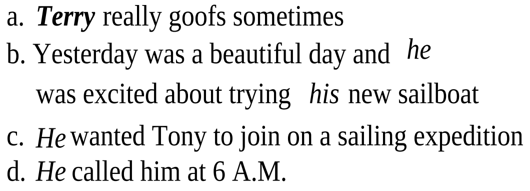
We assume a discourse to be a collection of sub-topics which contribute to the main topic of the discourse. The utterances/sentences which speak about a subtopic are aggregated as a discourse segment (DS). These discourse segments together form the discourse. A Discourse segment serves a purpose called discourse segment purpose (type of contribution to the topic of the discourse). We structure the discourse based on the above assumption by relating sentences which speak about the same topic to form discourse segments. Presently, We define the relationship between utterances/sentences within a discourse segment and the relationship between discourse segments as *dominance*. The relationship specifies the type of contribution to main topic which we refine as a part of future work.

In order to relate sentences which speak about the same topic we use surface information of a sentence and the evidences of cohesion. Two types of cohesion namely *coreference* and *lexical* cohesion prove to be vital evidence to find sentences which speak about a same topic. Coreference expressions refer to a previously introduced entity (center). So the sentence with a reference expression referring to a center in another sentence are said to discuss about the same topic (center). Lexical Cohesion is repetition of the same lexeme or a lexeme which is semantically similar (Hyponyms). Sentences which have similar surface information are also said to discuss about the same topic.

In this section we describe our model of parsing a discourse. We process the given discourse in three phases to progressively build a discourse tree structure similar to rhetorical tree structure[12] ,with one relation type *Dominance*[7], and maintaining the intentional, lexical structure of AST. In the first phase, we use a co-reference module to extract reference expressions (expression which refers to an Object, a Noun Phrase (NP) or a Prepositional Phrase (PP)) and their centers (the Object to which the reference expression refers to). We relate a sentence containing a reference expression to the sentence with the corresponding center as they speak about the same "Object" (topic). In the second phase we relate the remaining sentences using surface information and evidences of lexical cohesion. In the third phase we form discourse segments and find relations among them to generate a tree similar to RST. We now proceed with the description of the system.

2.1 First Phase

In the first phase, we use the output of the co-reference module (LingPipe)[1] to create a collection of trees from sentences. Given a discourse text, a co-reference module extracts reference expressions and marks their corresponding centers. For example, in Figure 1 the reference expression *he* refers to *Terry*, the center.



a. ***Terry*** really goofs sometimes

b. Yesterday was a beautiful day and *he* was excited about trying *his* new sailboat

c. *He* wanted Tony to join on a sailing expedition

d. *He* called him at 6 A.M.

Fig. 1: *Sample text with centers in bold italics, reference expressions in italics*

From this output of co-reference module we construct trees as follow: at the beginning, we consider each sentence to be a single node tree. Next, we find a parent to each of the sentences using the reference expressions as the basis. For this, consider a discourse of n sentences say S_1 to S_n. We make a sentence $S_i (i \neq 1)$ the child of S_{i-1} if any of the following hold

- There exists at least one reference expression in S_i with a center in S_{i-1}
- There exists at least one reference expression in S_i and one reference expression in S_{i-1} with a common center in some sentence $S_k, 1 \leq k < i-1$

At the end of this phase we have a collection of trees, each node with at most one child. Note that there is a possibility that a sentence, say S_i may not have a reference expression or may not have a center referred to by any reference expression. In this case, the sentence would remain a single-node tree. It is apparent from our construction that every tree now contains a chain of continuous sentences, each tree can be considered a partially built discourse segment. In the next phase, we merge these trees together into a single tree on the basis of lexical cohesion and surface information.

2.2 Second Phase

Suppose after the first phase we have a collection of z trees, T_1 to T_z. We maintain a final tree T, initialised to T_1, into which we merge all the remaining trees. This merging is done using lexical cohesion and surface information (similarity of content) between nodes (sentences) of candidate trees.

For merging $T_i, i \neq 1$ with T, consider the root sentence of T_i, say S_j. Extract all the Noun Phrases (NPs) and Prepositional Phrases (PPs) of S_j using a parser. We use the standard Stanford parser [8] for parsing. These NPs and PPs constitute *Surface Information.* We exploit the property of lexical cohesion in a discourse by assuming that the NPs of *Subject* semantic role and *Direct Object* semantic role have evidences of lexical cohesion. Therefore, out of all NPs and PPs of S_j, we consider the NPs with *Subject* semantic role (*NPSubject*) and *Direct Object* semantic role (*NPObject*) of the main verb. If the main verb is intransitive, we consider only the NP with *Subject* semantic role (*NPSubject*). For merging T_i into T, we find the most probable parent of T_i by comparing with the *NPSubject* and *NPObject* (if any) of S_j and the *Surface Information* of S_{j-1}, which exists in T, and all the ancestors of S_{j-1}. Our algorithm for merging T_i with T is as follows:

- Consider a set U = U_1, U_2,... U_k, U_1 the surface information of S_{j-1}, U_2 to U_k the surface information of ancestors of S_{j-1}, U_k the surface information root of T.
- Calculate the similarity between U_1 with *NPSubject* and *NPObject* (if any) of S_j, the process of similarity calculation being discussed later in the paper.
- For U_1 pick that entity which has maximum similarity (either with *NPSubject* or with *NPObject*) and define that entity as the topic between S_j and U_1.
- Repeat for all U_is and find the maximum of maximum similarities.
- If the maximum is unique and belongs to a node say S of T then we make S_j the child of S and thus merge T_i into T.
- If there is more than one occurrence of the maximum, then a decision based on lexical cohesion is not possible.
  – Inorder to find a most probable parent among these maxima, we prioritise the topics based on the importance of that topic. For this purpose, we use TextRank [13], which gives a rank to each topic based on the knowledge from the text. It is a graph based ranking algorithm using Google's Page Rank [2] on text. Out of the maxima, whichever topic has the highest priority, we choose the corresponding sentence as the parent of S_j.
  – If there exists a clash of priorities in this level also, then we resort to sentence similarities.

After calculating the similarity between U_i with *NPSubject* and *NPObject*, to avoid irrelevant entities, we pick only those entities/topics which have similarity value above a specific threshold (for better results $>$ 0.9). If there is no topic which has value greater than the threshold, then we consider sentence similarities. While considering sentence similarities, out of all the ancestors of S_{j-1} (including itself), the sentence (node) most similar to S_j is made the parent of S_j. For example in Figure 2, T_1 and T_2 are the first level trees formed from five sentences S_1 to S_5. We initialise T_1 to T and for merging T_2 into T, we calculate the similarity value between S_4 with S_3 and all its ancestors. The final similarity measures are shown in the figure, S_4 is made the child of S_2 as it has the maximum similarity measure.

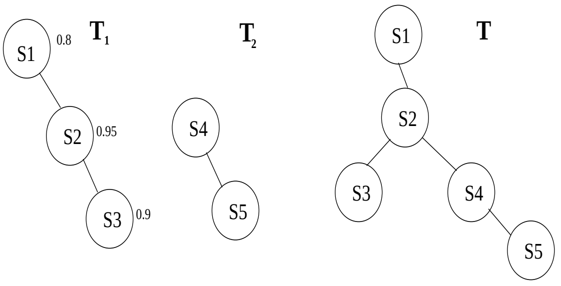
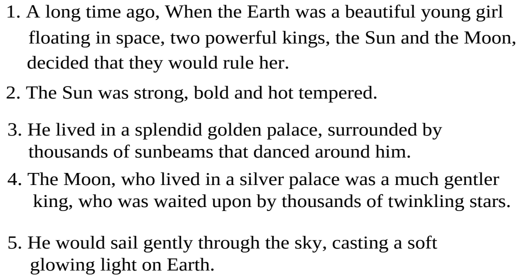


Fig. 2: *Tree merging of second phase*

The similarity of our construction to that of Attentional State Theory can now be seen easily. In Attentional State theory, each sentence starts a discourse segment or continues a discourse segment or ends a discourse segment. In our case, each time a sentence S_j attaches itself as the child of S_k, it figuratively means that it either continues the topic of S_k or ends the topic of S_k. This also means that S_j ends all the topics of siblings of S_k. Therefore, from now on for a sentence S_{j+1}, there is no need to consider all the nodes of T. It would suffice to consider S_j and its ancestors since it cannot continue an already ended topic. It is apparent that we are refining the partial discourse segments constructed in first phase during the second phase. In the next section, we describe the procedure to find similarity between two sentences.

2.2.1 Similarity between sentences

Given two sentences S_1 and S_2, we calculate the similarity measure between these two as follows: Extract the NPs and PPs of S_1 and S_2, identify the common nouns (NN) and proper nouns (NNP) from each NP and PP. Out of these filter out only those NNs and NNPs whose degree of importance (from TextRank) is above a threshold. For calculating the similarity between these two sentences, we use Dynamic Time Warping Algorithm (DTW) [11] on the filtered set of NNs and NNPs. Dynamic Time Warping Algorithm calculates the sentence similarities using dynamic programming, with a matrix whose rows correspond to the words of first sentence and columns correspond to the words of the second sentence, the similarity score between a word of one sentence and a word of another sentence being the elements of the matrix. In our case the rows correspond to the filtered NNs/NNPs of S_1, columns correspond to the filtered NNs/NNPs of S_2. For each grid element the following recurrence formula is defined



1. A long time ago, When the Earth was a beautiful young girl floating in space, two powerful kings, the Sun and the Moon, decided that they would rule her.
2. The Sun was strong, bold and hot tempered.
3. He lived in a splendid golden palace, surrounded by thousands of sunbeams that danced around him.
4. The Moon, who lived in a silver palace was a much gentler king, who was waited upon by thousands of twinkling stars.
5. He would sail gently through the sky, casting a soft glowing light on Earth.

Fig. 3: *Sample Text*

$$D_{dtw}(S_1, S_2) = f(m, n)$$

$$f(i,j) = d(S_{1_i}, S_{2_j}) + min \begin{cases} f(i-1,j) \\ f(i,j-1) \\ f(i-1,j-1) \end{cases}$$

$$f(0,0) = 0, f(i,0) = f(0,j) = \infty$$

$$i \in (0,m), j \in (0,n)$$

Where m = no. of rows, n = no. of columns, $d(a,b)$ the similarity measure between words a and b. After the execution of DTW, the normalized value at the last element of the last row gives the similarity measure between S_1 and S_2. The minimum this value, the maximum the similarity between the sentences. We now describe similarity calculation between a pair of words.

2.2.2 Similarity between words

Several similarity metrics have been proposed so far to calculate the similarity between a pair of words. In [10], the similarity is calculated using the formula

$$Sim_{lch} = -\log \frac{length}{2 * D}$$

Where length is the length of the shortest path between two concepts using node-counting, and D is the maximum depth of the taxonomy. According to [17], the similarity metric measures the depth of two given concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS), and combines these figures into a similarity score

$$Sim_{wup} = \frac{2 * depth(LCS)}{depth(concept1) + depth(concept2)}$$

In our paper, we define the similarity score of a pair of words as the average of Synonym similarity and Hypernym similarity. We calculate

$$Sim_w(w_1, w2) = \frac{e^{\alpha d} - 1}{e^{\alpha d} + e^{\beta l} - 2}$$

for Hypernym similarity [11]. α and β[1] are smoothing factors, l is the shortest path length between w_1 and w_2; d is the depth of subsumer in the hierarchy semantic nets extracted from WordNet [14]. For Synonym similarity the normalized value of the number of N-gram (N = 3) matchings between the synonyms of two words is taken.

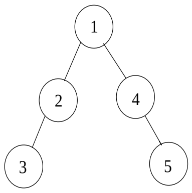
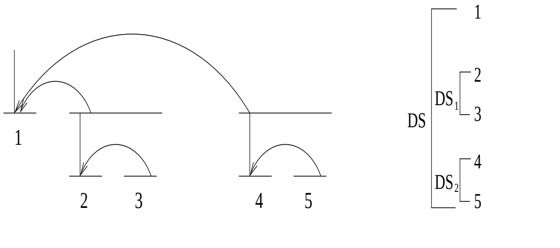


Fig. 4: *Final tree after the second phase*



a. Rhetorical Structure Tree **b. Discourse segments**

Fig. 6: *Resulting structures from Figure 4*

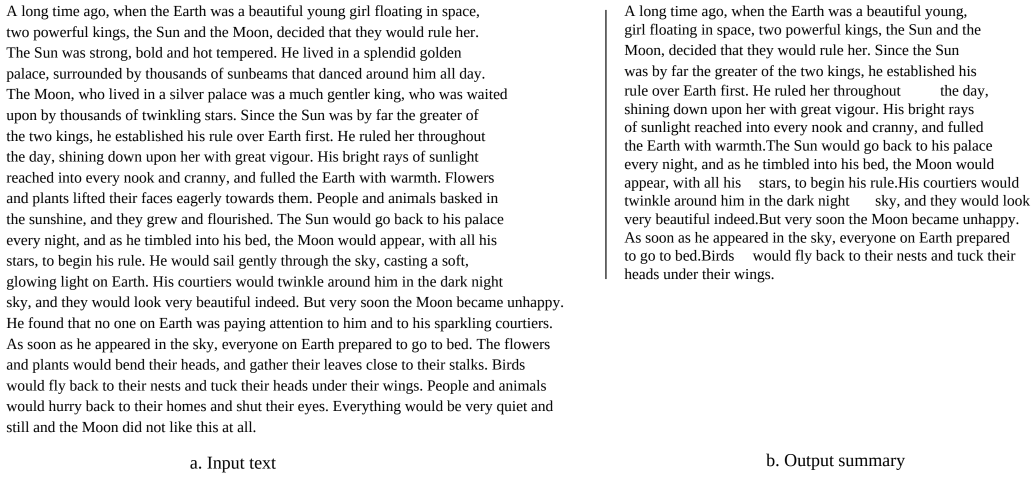
2.3 Third Phase

At this stage we have the discourse tree structure where each sentence (node) has a parent (except the root) and each node can have any number of children. We can interpret this tree as follows: The root node starts a new topic. If the root node has more than one children then each of the subtrees of the children in turn contribute to the root node ,but independently. Hence the subtrees rooted at these children can be viewed as separate discourse segments. If the root node has only one child then the root node and the child combined together start a topic. This can be performed at all levels of the tree and a hierarchical discourse segment structure can be built. Each node/sentence that starts a discourse segment has a *dominance* relation to the sentences of that segment.

To make our tree structure similar to that of rhetorical tree structure we identify the nucleus and satellite nodes from the hierarchical discourse segments. Each discourse segment is characterized by a nucleus which is recursively calculated from the sentence(s) starting that segment and the nuclei of the sub-discourse segments of that segment. The nucleus of a segment is the sentence with maximum sum of TextRank values of Noun Phrases (normalized) among the candidate sentences. The remaining sentences are satellites of this sentence.

We demonstrate the algorithm with an example. Figure 3 shows a sample text for which the final tree using the algorithm of second phase described above is shown in Figure 4. Now after the third phase, from the tree in Figure 4, we identify the discourse segments and build RST as shown in Figure 6

[1] $\alpha = 1.0$, $\beta = 0.45$



A long time ago, when the Earth was a beautiful young girl floating in space, two powerful kings, the Sun and the Moon, decided that they would rule her. The Sun was strong, bold and hot tempered. He lived in a splendid golden palace, surrounded by thousands of sunbeams that danced around him all day. The Moon, who lived in a silver palace was a much gentler king, who was waited upon by thousands of twinkling stars. Since the Sun was by far the greater of the two kings, he established his rule over Earth first. He ruled her throughout the day, shining down upon her with great vigour. His bright rays of sunlight reached into every nook and cranny, and fulled the Earth with warmth. Flowers and plants lifted their faces eagerly towards them. People and animals basked in the sunshine, and they grew and flourished. The Sun would go back to his palace every night, and as he timbled into his bed, the Moon would appear, with all his stars, to begin his rule. He would sail gently through the sky, casting a soft, glowing light on Earth. His courtiers would twinkle around him in the dark night sky, and they would look very beautiful indeed. But very soon the Moon became unhappy. He found that no one on Earth was paying attention to him and to his sparkling courtiers. As soon as he appeared in the sky, everyone on Earth prepared to go to bed. The flowers and plants would bend their heads, and gather their leaves close to their stalks. Birds would fly back to their nests and tuck their heads under their wings. People and animals would hurry back to their homes and shut their eyes. Everything would be very quiet and still and the Moon did not like this at all.

a. Input text

A long time ago, when the Earth was a beautiful young, girl floating in space, two powerful kings, the Sun and the Moon, decided that they would rule her. Since the Sun was by far the greater of the two kings, he established his rule over Earth first. He ruled her throughout the day, shining down upon her with great vigour. His bright rays of sunlight reached into every nook and cranny, and fulled the Earth with warmth.The Sun would go back to his palace every night, and as he timbled into his bed, the Moon would appear, with all his stars, to begin his rule.His courtiers would twinkle around him in the dark night sky, and they would look very beautiful indeed.But very soon the Moon became unhappy. As soon as he appeared in the sky, everyone on Earth prepared to go to bed.Birds would fly back to their nests and tuck their heads under their wings.

b. Output summary

Fig. 5: *Summarization*

3 Summarization and Results

We apply our model of discourse parsing for text summarization. From a given text, we the nuclei of discourse segments upto a certain threshold level which depends on the compression factor.

Figure 5 shows a sample English text and the result of summarization. We plan to test our summarization on a standard corpus. We can improve this model using better summarization techniques on the discourse tree.

4 Future Work

Our first task would be to perform a thorough evalution of the current model. Presently our model supports single relation type, *dominance*. We aim at accommodating more relation types into our model. Relation types can be extracted by formulating linguistic rules on cue-phrases such as *therefore, whereas, On the other hand* and so on which would increase the quality of discourse tree further. We can also extend our work to Indian languages such as Hindi which would be a challenging task due to inadequate linguistic resources.

This paper is an outline of system to be developed. The work is still in progress and even though improvements need to be done, we can see that the summary resulting from our rough model is considerably meaningful which in turn signifies the efficacy of our idea.

References

[1] Alias-i. `http://alias-i.com/lingpipe`, 2008.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine, 1998.

[3] S. Corston-Oliver and S. H. Corston-oliver. Beyond string matching and cue phrases: Improving efficiency and coverage in discourse analysis. In *The AAAI Spring Symposium on Intelligent Text Summarization*, pages 9–15, 1998.

[4] D. Cristea. An incremental discourse parser architecture. *Lecture Notes in Computer Science*, 1835:162–174, 2000.

[5] D. Cristea, N. Ide, and L. Romary. Veins theory: a model of global discourse cohesion and coherence. In *Proceedings of the 17th international conference on Computational linguistics*, pages 281–285, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[6] K. Forbes, E. Miltsakaki, R. Prasad, A. Sarkar, A. Joshi, B. Webber, A. Joshi, and B. Webber. D-ltag system: Discourse parsing with a lexicalized tree adjoining grammar. *Journal of Logic, Language and Information*, 12:261–279, 2002.

[7] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[8] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[9] S. Kurohashi and M. Nagao. Automatic detection of discourse structure by checking surface information in sentences. In *Proceedings of the 15th conference on Computational linguistics*, pages 1123–1127, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[10] C. Leacock and M. Chodorow. Combining local context with wordnet similarity for word sense identification. In *WordNet: A Lexical Reference System and its Application*. MIT Press, 1998.

[11] X. Liu, Y. Zhou, and R. Zheng. Sentence similarity based on dynamic time warping. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, pages 250–256, Washington, DC, USA, 2007. IEEE Computer Society.

[12] W. C. Mann and S. A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[13] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004.

[14] G. Miller. Wordnet: a lexical database for english, 1995.

[15] L. Polanyi, C. Culy, M. van den Berg, G. L. Thione, and D. Ahn. A rule based approach to discourse parsing. In M. Strube and C. Sidner, editors, *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 108–117, Cambridge, Massachusetts, USA, April 30 - May 1 2004. Association for Computational Linguistics.

[16] R. Soricut and D. Marcu. Sentence level discourse parsing using syntactic and lexical information. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 149–156, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[17] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico, 1994.

Author Index