Coling 2008

22nd International Conference on Computational Linguistics

Proceedings of the workshop on Knowledge and Reasoning for Answering Questions

Workshop chairs:
Marie-Francine Moens, Patrick Saint-Dizier

23 August 2008
Manchester, UK

Order copies of this and other Coling proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-905593-53-8

Design by Chimney Design, Brighton, UK
Production and manufacture by One Digital, Brighton, UK

Introduction

Welcome to the ACL Workshop on Unresolved Matters. We received 17 submissions, and due to a rigerous review process, we rejected 16.

Table of Contents

Conference Programme

Saturday, 23 August 2008

09:00–10:00 *Semantic Chunk Annotation for complex questions using Conditional Random Field*
Shixi Fan, Yaoyun Zhang, Wing W. Y. Ng, Xuan Wang and Xiaolong Wang

10:00–10:30 *Context Inducing Nouns*
Charlotte Price, Valeria de Paiva and Tracy Holloway King

10.30–11.00 Break

11.00–12.00 Invited Talk

12.00–13.30 Lunch

13:30–14:00 *Know-Why Extraction from Textual Data for Supporting What Questions*
Chaveevan Pechsiri, Phunthara Sroison and J. Janviriyasopak

14:00–14:30 *Context Modelling for IQA: the Role of Tasks and Entities*
Raffaella Bernardi and Manuel Kirschner

14:30–15:00 *Personalized, Interactive Question Answering on the Web*
Silvia Quarteroni

15:00–15:30 *Creating and Querying a Domain dependent Know-How Knowledge Base of Advices and Warnings*
Lionel Fontan and Patrick Saint-Dizier

15.30–16.00 Break

16.00–17.30 Panel: Multimedia question answering

Semantic Chunk Annotation for complex questions using Conditional Random Field

Shixi Fan
Department of computer science
Harbin Institute of Technology
Shenzhen Graduate School,
Shenzhen, 518055, china
fanshixi@hit.edu.cn

Yaoyun Zhang
Department of computer science
Harbin Institute of Technology
Shenzhen Graduate School,
Shenzhen, 518055, china
Xiaoni5122@gmail.com

Wing W. Y. Ng
Department of computer science
Harbin Institute of Technology
Shenzhen Graduate School,
Shenzhen, 518055, china
wing@hitsz.edu.cn

Xuan Wang
Department of computer science
Harbin Institute of Technology
Shenzhen Graduate School,
Shenzhen, 518055, china
wangxuan@insun.hit.edu.cn

Xiaolong Wang
Department of computer science
Harbin Institute of Technology
Shenzhen Graduate School,
Shenzhen, 518055, china
wangxl@insun.hit.edu.cn

Abstract

This paper presents a CRF (Conditional Random Field) model for Semantic Chunk Annotation in a Chinese Question and Answering System (SCACQA). The model was derived from a corpus of real world questions, which are collected from some discussion groups on the Internet. The questions are supposed to be answered by other people, so some of the questions are very complex. Mutual information was adopted for feature selection. The training data collection consists of 14000 sentences and the testing data collection consists of 4000 sentences. The result shows an F-score of 93.07%.

1 Introduction

1.1 Introduction of Q&A System

Automated question answering has been a hot topic of research and development since the earliest AI applications (A.M. Turing, 1950). Since then there has been a continual interest in processing knowledge and retrieving it efficiently to users automatically. The end of the 1980s saw a boost in information retrieval technologies and applications, with an unprecedented growth in the amount of digital information available, an explosion of growth in the use of computers for communications, and the increasing number of users that have access to all this information (Diego Moll′and Jose′Luis Vicedo, 2007). Search engines such as Google, Yahoo, Baidu and etc have made a great success for people's information need.

Anyhow, search engines are keywords-based which can only return links of relevant web pages, failing to provide a friendly user-interface with queries expressed in natural language sentences or questions, or to return precise answers to users. Especially from the end of the 1990s, as

Coling 2008: Proceedings of the workshop on Knowledge and Reasoning for Answering Questions, pages 1–8
Manchester, August 2008

information retrieval technologies and methodologies became mature and grew more slowly in pace, automated question answering(Q&A) systems which accept questions in free natural language formations and return exactly the answer or a short paragraph containing relevant information has become an urgent necessity. Major international evaluations such as TREC, CLEF and NTCIR have attracted the participation of many powerful systems.

The architecture of a Q&A system generally includes three modules: question processing, candidate answer/document retrieval, and answer extraction and re-ranking.

1.2 Introduction of Question Analyzing

Question Analyzing, as the premise and foundation of the latter two modules, is of paramount importance to the integrated performance of a Q&A system. The reason is quite intuitive: a question contains all the information to retrieve the corresponding answer. Misinterpretation or too much loss of information during the processing will inevitably lead to poor precision of the system.

The early research efforts and evaluations in Q&A were focused mainly on factoid questions asking for named entities, such as time, numbers, and locations and so on. The questions in the test corpus of TREC and other organizations are also in short and simple form. Complex hierarchy in question types (Dragomir Radev et al, 2001), question templates (Min-Yuh Day et al, 2005), question parsing (Ulf Hermjakob, 2001) and various machine learning methods (Dell Zhang and Wee Sun Lee, 2003)are used for factoid question analysis, aiming to find what named entity is asked in the question. There are some questions which are very complicated or even need domain restricted knowledge and reasoning technique. Automatic Q&A system can not deal with such questions with current technique.

In china, there is a new kind of web based Q&A system which is a special kind of discussion group. Unlike common discussion group, in the web based Q&A system one user posts a question, other users can give answers to it. It is found that at least 50% percent questions (Valentin Jijkoun and Maarten de Rijke, 2005)posted by users are non-factoid and surely more complicated both in question pattern and information need than those questions in the test set of TREC and other FAQ. An example is as follows:

{我想开专卖店工商如何交费手续如何办理}
(I want to open a special sell shop, how to pay the tax, and what is the procedure of it)

This kind of Q&A system can complement the search engines effectively. As the best search engines in china, Baidu open the Baidu Knowledge[2] Q&A system from 2003, and now it has more than 29 million question-answer pairs. There are also many other systems of this kind such as Google Groups, Yahoo Answers and Sina Knowledge[3]. This kind of system is a big question-answer pair database which can be treated as a FAQ database. How to search from the database and how to analyze the questions in the database needs new methods and techniques.

More deeper and precise capture of the semantics in those complex questions is required. This phenomenon has also been noticed by some researchers and organizations. The spotlight gradually shifted to the processing and semantic understanding of complex questions. From 2006, TREC launched a new annually evaluation CIQ&A (complex, interactive Question Answering), aiming to promote the development of interactive systems capable of addressing complex information needs. The targets of national programs AQUAINT and QUETAL are all at new interface and new enhancements to current state-of-the-art Q&A systems to handle more complex inputs and situations.

A few researchers and institutions serve as pioneers in complex questions study. Different technologies, such as definitions of different sets of question types, templates and sentence patterns (Noriko Tomuro, 2003) (Hyo-Jung Oh et al, 2005) machine learning methods (Radu Soricut and Eric Brill, 2004), language translation model (Jiwoon Jeon, W et al, 2005), composition of information needs of the complex question (Sanda Harabagiu et al, 2006) and so on, have been experimented on the processing of complex question, gearing the acquired information to the facility of other Q&A modules.

Several major problems faced now by researcher of complex questions are stated as follow:

First: Unlike factoid questions, it is very difficult to define a comprehensive type hierarchy for complex questions. Different domains under research may require definitions of different sets of question types, as shown in (Hyo-Jung Oh et al, 2005). Especially, the types of certain ques-

[2] http://zhidao.baidu.com/
[3] http://iask.sina.com.cn/

tions are ambiguous and hard to identify. For example:

网上炒股的方法是什么？

（how to play stock market ?）

This question type can be treated as definition, procedure or entity.

Second: Lack of recognition of different semantic chunks and the relations between them. FAQFinder (Radu Soricut and Eric Brill, 2004) also used semantic measure to credit the similarity between different questions. Nevertheless, the question similarity is only a simple summation of the semantic similarity between words from the two question sentences. Question pattern are very useful and easy to implement, as justified by previous work. However, just like the problem with question types, question patterns have limitation on the coverage of all the variations of complex question formation. Currently, after the question processing step in most systems, the semantic meaning of large part of complex questions still remain vague. Besides, confining user's input only within the selection of provided pattern may lead to unfriendly and unwelcome user interface.

(Ingrid Zukerman and Eric Horvitz, 2001) used decision tree to model and recognize the information need, question and answer coverage, topic, focus and restrictions of a question. Although features employed in the experiments were described in detail, no selection process of those feature, or comparison between them was mentioned.

This paper presents a general method for Chinese question analyzing. Our goal is to annotate the semantic chunks for the question automatically.

2 Semantic Chunk Annotation

Chinese language differs a lot from English in many aspects. Mature methodologies and features well-justified in English Q&A systems are valuable sources of reference, but no direct copy is possible.

The Ask-Answer system[4] is a Chinese online Q&A system where people can ask and answer questions like other web based Q&A system. The characteristic of this system is that it can give the answer automatically by searching from the asked question database when a new question is presented by people. The architecture of the automatically answer system is shown in figure 1. The system contains a list of question-answer pairs on particular subject. When users input a question from the web pages, the question is submitted to the system and then question-answer pair is returned by searching from the questions asked before. The system includes four main parts: question pre-processing, question analyzing, searching and answer getting.

The question pre-processing part will segment the input questions into words, label POS tags for every word. Sometimes people ask two or more questions at one time, the questions should be made into simple forms by conjunctive structure detection. The question analyzing program will find out the question type, topic, focus and etc. The answer getting part will get the answer by computing the similarity between the input question and the questions asked before. The question analyzing part annotates the semantic chunks for the question. So that the question can be mapped into semantic space and the question similarity can be computed semantically. The Semantic chunk annotation is the most important part of the system.

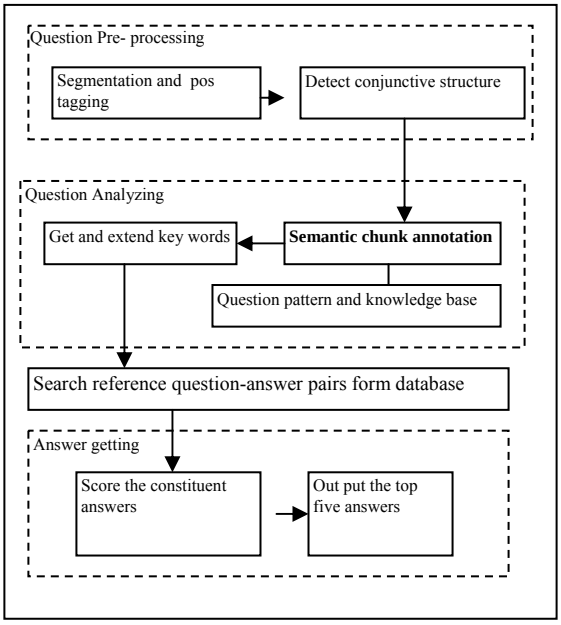

Figure 1 the architecture of the automatically answer system

Currently, no work has been reported yet on the question semantic chunk annotation in Chinese. The prosperity of major on-line discussion groups provides an abundant ready corpus for question answering research. Using questions collected from on-line discussion groups; we make a deep research on semantic meanings and build a question semantic chunk annotation model based on Conditional Random Field.

Five types of semantic chunks were defined: Topic, Focus, Restriction, Rubbish information and Interrogative information. The topic of a

[4] http://haitianyuan.com/qa

question which is the topic or subject asked is the most important semantic chunk. The focus of a question is the asking point of the question. The restriction information can restrict the question's information need and the answers. The rubbish information is those words in the question that has no semantic meanings for the question. Interrogative information is a semantic tag set which corresponds to the question type. The interrogative information includes interrogative words, some special verbs and nouns words and all these words together determine the question type. The semantic chunk information is shown in table 1.

Semantic chunk tag	Abbreviation	Meaning
Topic	T	The question subject
Focus	F	The additional information of topic
Restrict	Re	Such as Time restriction and location restriction
Rubbish information	Ru	Words no meaning for the question
Other	O	other information without semantic meaning
The following is interrogative information		
Quantity	Wqua	
Description	Wdes	The answer need description
Yes/No	Wyes	The answer should be yes or no
List	Wlis	The answer should be a list of entity
Definition	Wdef	The answer is the definition of topic
Location	Wloc	The answer is location
Reason	Wrea	The answer can explain the question
Contrast	Wcon	The answer is the comparison of the items proposed in the question
People	Wwho	The answer is about the people's information
Choice	Wcho	The answer is one of the choice proposed in the question
Time	Wtim	The answer is the data or time length about the event in the question
Entity	Went	The answer is the attribute of the topic.

Table 1: Semantic chunks

An annotation example question is as follows:

银行交易类型 CMD 是什么意思？
What is the meaning of bank transaction CMD?

This question can be annotated as follows:

{银行/nz 交易/nz 类型/n CMD/nx}/T
{是/v 什么/r 意思/n}/Wdef ？/w

This kind of annotation is not convenient for CRF model, so the tags were transfer into the B I O form. (Shown as follows)

银行/nz/B-T 交易/nz/I-T 类型/n/I-T
CMD/nx/I-T 是/v/B-Wdef 什么/r/I-Wdef
意思/n/I-Wdef ？/w/O

Then the Semantic chunk annotation can be treated as a sequence tag problem.

3 Semantic Chunk Annotation model

3.1 Overview of the CRF model

The conditional random field (CRF) is a discriminative probabilistic model proposed by John Lafferty, et al (2001) to overcome the long-range dependencies problems associated with generative models. CRF was originally designed to label and segment sequences of observations, but can be used more generally. Let *X*, *Y* be random variables over observed data sequences and corresponding label sequences, respectively. For simplicity of descriptions, we assume that the random variable sequences *X* and *Y* have the same length, and use $x = [x_1, x_2 \ldots\ldots x_m]$ and $y = [y_1, y_2 \ldots\ldots y_m]$ to represent instances of *X* and *Y*, respectively. CRF defines the conditional probability distribution $P(Y|X)$ of label sequences given observation sequences as follows

$$P_\lambda(Y \mid X) = \frac{1}{Z_\lambda(X)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(X, Y)\right) \quad (1)$$

Where $Z_\lambda(X)$ is the normalizing factor that ensures equation 2.

$$\sum_y P_\lambda(y \mid x) = 1 \quad (2)$$

In equation 2 the λ_i is a model parameter and $f_i(X, Y)$ is a feature function (often binary-valued) that becomes positive (one for binary-valued feature function) when *X* contains a certain feature in a certain position and *Y* takes a certain label, and becomes zero otherwise. Unlike Maximum Entropy model which use single normalization constant to yield a joint distribution, CRFs use the observation-dependent normalization $Z_\lambda(X)$ for conditional distributions. So CRFs can avoid the label biased problem. Given a set of training data $T = \{(x_k, y_k), k = 1,2 \ldots . n\}$

With an empirical distribution $\tilde{P}(X, Y)$, CRF

determines the model parameters $\lambda = \{\lambda_i\}$ by maximizing the log-likelihood of the training set

$$\Gamma(P_\lambda) = \sum_{k=1}^{N} \log P_\lambda(y_k \mid x_k) \propto \sum_{x,y} \tilde{P}(x,y) \log P_\lambda(y \mid x) \quad (3)$$

3.2 Features for the model

The following features, which are used for training the CRF model, are selected according to the empirical observation and some semantic meanings. These features are listed in the following table.

Feature type index	Feature type name
1	Current word
2	Current POS tag
3	Pre-1 word POS tag
4	Pre-2 word POS tag
5	Post -1 word POS tag
6	Post -2 word POS tag
7	Question pattern
8	Question type
9	Is pattern key word
10	Pattern tag

Table 2: the Features for the model

Current word:
The current word should be considered when adding semantic tag for it. But there are too many words in Chinese language and only part of them will contribute to the performance, a set of words was selected. The ***word set*** includes segment note and some key words such as time key word and rubbish key word. When the current word is in the ***word set*** the current word feature is the current word itself, and null on the other hand.

Current POS tag:
Current POS tag is the part of speech tag for the current word.

Pre-1 word POS tag:
Pre- 1 word POS tag is the POS tag of the first word before the labeling word in the sentence. If the Pre-1 word does not exit (the current is the first word in the sentence), the Pre- 1 word POS tag is set to null.

Pre-2 word POS tag:
Pre- 2 word POS tag is the POS tag of the second word before the labeling word in the sentence. If the Pre-2 word does not exit, the Pre- 2 word POS tag is set to null.

Post -1 word POS tag:
Post - 1 word POS tag is the POS tag of the first word after the labeling word in the sentence. If the Post -1 word does not exit (the current is the first word in the sentence), the Post - 1 word POS tag is set to null.

Post -2 word POS tag:
Post - 2 word POS tag is the POS tag of the second word after the labeling word in the sentence. If the Post-2 word does not exit, the Pre- 2 word POS tag is set to null.

Question pattern:
Question pattern which is associated with question type, can locate question topic, question focus by surface string matching. For example, (where is <topic>). The patterns are extracted from the training data automatically. When a pattern is matched, it is treated as a feature. There are 1083 question patterns collected manually.

Question type:
Question type is an important feature for question analyzing. The question patterns have the ability of deciding the question type. If there is no question pattern matching the question, the question type is defined by a decision tree algorithm.

Is pattern key word:
For each question pattern, there are some key words. When the current word belongs to the pattern key word this feature is set to "yes", else it is set to "no".

Pattern tag:
When a pattern is matched, the topic, focus and restriction can be identified by the pattern. We can give out the tags for the question and the tags are treated as features. If there is no pattern is matched, the feature is set to null.

4 Feature Selection experiment

Feature selection is important in classifying systems such as neural networks (NNs), Maximum Entropy, Conditional Random Field and etc. The problem of feature selection has been tackled by many researchers. Principal component analysis (PCA) method and Rough Set Method are often used for feature selection. Recent years, mutual information has received more attention for feature selection problem.

According to the information theory, the uncertainty of a random variable X can be measured by its entropy $H(X)$. For a classifying problem, there are class label set represented by C and feature set represented by F. The conditional entropy $H(C \mid F)$ measures the uncertainty about

C when F is known, and the Mutual information I(C, F) is defined as:

$$I(C;F) = H(\mathrm{C}) - H(\mathrm{C} \mid \mathrm{F}) \quad (4)$$

The feature set is known; so that the objective of training the model is to minimize the conditional entropy $H(C \mid F)$ equally maximize the mutual information $I(C;F)$. In the feature set F, some features are irrelevant or redundant. So that the goal of a feature selection problem is to find a feature S ($S \subset F$), which achieve the higher values of $I(C;F)$. The set S is a subset of F and its size should be as small as possible. There are some algorithms for feature selection problem. The ideal greedy selection algorithm using mutual information is realized as follows (Nojun Kwak and Chong-Ho Choi, 2002):

Input: S- an empty set

F- The selected feature set

Output: a small reduced feature set S which is equivalent to F

Step 1: calculate the MI with the Class set C , $\forall f_i \in F$, compute $I(C; f_i)$

Step 2: select the feature that maximizes $I(C; f_i)$, set $F \leftarrow F \setminus \{f_i\}, S \leftarrow \{f_i\}$

Step 3: repeat until desired number of features are selected.

1) Calculate the MI with the Class set C and S, $\forall f_i \in F$, compute $I(C;S, f_i)$
2) Select the feature that maximizes $I(C;S, f_i)$, set $F \leftarrow F \setminus \{f_i\}, S \leftarrow \{f_i\}$

Step 4: Output the set S that contains the selected features

To calculate MI the PDFs (Probability Distribution Functions) are required. When features and classing types are dispersing, the probability can be calculated statistically. In our system, the PDFs are got from the training corpus statistically.

The training corpus contains 14000 sentences. The training corpus was divided into 10 parts, with each part 1400 sentences. And each part is divided into working set and checking set. The working set, which contains 90% percent data, was used to select feature by MI algorithm. The checking set, which contains 10% percent data, was used to test the performance of the selected feature sequence. When the feature sequence was selected by the MI algorithm, a sequence of CRF models was trained by adding one feature at each time. The checking data was used to test the performance of these models.

	The open test result									
Selected feature sequence	1	2	3	4	5	6	7	8	9	10
7, 10, 3, 1, 5, 2, 4, 6, 8，9	0.5104	0.8764	0.8864	0.8918	0.8925	0.8977	0.8992	0.9023	0.9025	0.9018
7, 10, 1, 3, 5, 2, 4，6，8，9	0.5241	0.8775	0.8822	0.8911	0.8926	0.8956	0.8967	0.9010	0.9005	0.9007
7, 10, 1, 3, 5, 2, 4, 6，8，9	0.5090	0.8691	0.8748	0.8851	0.8852	0.8914	0.8929	0.8955	0.8955	0.8949
7, 10, 1, 3, 5, 2, 4, 6，9，8	0.5157	0.8769	0.8823	0.8913	0.8925	0.8978	0.8985	0.9017	0.9018	0.9010
7, 10, 1, 3, 5, 2, 4, 6，8，9	0.5144	0.8821	0.8856	0.8921	0.8931	0.8972	0.8981	0.9010	0.9009	0.9007
7, 10, 3, 1, 5, 2, 4，6，8，9	0.5086	0.8795	0.8876	0.8914	0.8919	0.8960	0.8967	0.9016	0.9013	0.9011
7, 10, 1, 3, 5, 2, 4, 6, 8, 9	0.5202	0.8811	0.8850	0.8920	0.8931	0.8977	0.8980	0.9015	0.9013	0.9009
7, 10, 1, 3, 5, 2, 4, 6，8，9	0.5015	0.8858	0.8879	0.8948	0.8942	0.8998	0.8992	0.9033	0.9027	0.9023
7, 10, 1, 3, 5, 2, 4, 6，8，9	0.5179	0.8806	0.8805	0.8898	0.8908	0.8954	0.8958	0.8982	0.8982	0.8986
7, 10, 1, 3, 5, 2, 4, 6, 8，9	0.5153	0.8921	0.8931	0.9006	0.9012	0.9041	0.9039	0.9071	0.9068	0.9067

Table 3: the feature selection result and the test result

In table 3, each row contains data corresponding to one part of the training corpus so there are ten rows with data in the table. The third row corresponds to the first part and the last row corresponds to the tenth part. There are eleven columns in the table, the first columns is the features sequence selected by the mutual information algorithm for each part. The second column is the open test result with the first feature in the feature sequence. The third column is the open test result with the first two features in the feature sequence and so on. From the table, it is

clear that the feature 7(Question pattern) and 10(Pattern tag) are very important, while the feature 8(Question type) and 9(Is pattern key word) are not necessary. The explanation about this phenomenon is that the "pattern key word" and "Question type" information can be covered by the Question patterns. So feature 8 and 9 are not used in the Conditional Random Field model.

5 Semantic Chunk Annotation Experiment

The test and training data used in our system are collected from the website (Baidu knowledge and the Ask-Answer system), where people proposed questions and answers. The training data consists of 14000 and the test data consists of 4000 sentences. The data set consists of word tokens, POS and semantic chunk tags. The POS and semantic tags are assigned to each word tokens.

The performance is measured with three rates: precision (Pre), recall (Rec) and F-score (F1).

$$Pre = Match/Model \quad (5)$$

$$Rec = Match/Manual \quad (6)$$

$$F1 = 2*Pre*Rec/(Pre+Rec) \quad (7)$$

Match is the count of the tags that was predicted right. Model is the count of the tags that was predicted by the model. Manual is the count of the tags that was labeled manually.

Table 4 shows the performance of annotation of different semantic chunk types. The first column is the semantic chunk tag. The last three columns are precision, recall and F1 value of the semantic chunk performance, respectively.

Label	Manual	Model	Match	Pre.()	Rec.()	F1
B-T，I-T	17061，78462	16327，80488	14825，76461	90.80，95.00	86.89，97.45	88.80，96.21
B-F，I-F	5072，13029	5079，13583	4657，12259	91.69，90.25	91.82，94.09	91.75，92.13
B-Ru，I-Ru	775，30	11，0	2，0	18.18，0.00	0.26，0.00	0.51，0.00
O	8354	8459	6676	78.92	79.91	79.41
B-Wqua，I-Wqua	1363，934	1327，1028	1298，881	97.81，85.70	95.23，94.33	96.51，89.81
B-Wyes，I-Wyes	5669，1162	5702，1098	5550，1083	97.33，98.63	97.90，93.20	97.62，95.84
B-Wdes，I-Wdes	2907，278	2855，185	2779，184	97.34，99.46	95.60，66.19	96.46，79.48
B-Wlis，I-Wlis	603，257	563，248	560，248	99.47，100	92.87，96.50	96.05，98.22
B-Wdef，I-Wdef	1420，1813	1430，1878	1280，1695	89.51，90.26	90.14，93.49	89.82，91.85
B-Wloc，I-Wloc	683，431	665，395	661，392	99.40，99.24	96.78，90.95	98.07，94.92
B-Wrea，I-Wrea	902，159	873，83	843，82	96.56，98.80	93.46，51.57	94.99，67.77
B-Wcon，I-Wcon	552，317	515，344	503，291	97.67，84.59	91.12，91.80	94.28，88.05
B-Wwho，I-Wwho	420，364	357，350	348，336	97.48，96.00	82.86，92.31	89.58，94.12
B-Wcho，I-Wcho	857，85	738，0	686，0	92.95，0.00	80.05，0.00	86.02，0.00
B-Wtim，I-Wtim	408，427	401，419	355，380	88.53，90.69	87.01，88.99	87.76，89.83
B-Went，I-Went	284，150	95，81	93，80	97.89，98.77	32.75，53.33	49.08，69.26
Avg	145577	145577	135488	93.07	93.07	93.07

Table 4: the performance of different semantic chunk

The semantic chunk type of "Topic" and "Focus" can be annotated well. Topic and focus semantic chunks have a large percentage in all the semantic chunks and they are important for question analyzing. So the result is really good for the whole Q&A system.

As for "Rubbish" semantic chunk, it only has 0.51 and 0.0 F1 measure for B-Ru and I-Ru. One reason is lacking enough training examples, for there are only 1031 occurrences in the training data. Another reason is sometimes restriction is complex.

6 Conclusion and future work

This paper present a new method for Chinese question analyzing based on CRF. The features are selected by using mutual information algorithm. The selected features work effectively for the CRF model. The experiments on the test data set achieve 93.07% in F1 measure. In the future, new features should be discovered and new methods will be used.

Acknowledgment

This work is supported by Major Program of National Natural Science Foundation of China (No.60435020 and No. 90612005) and the High Technology Research and Development Program of China (2006AA01Z197).

References

A.M. Turing. 1950. *Computing Machinery and Intelligence*. Mind, 236 (59): 433~460.

Diego Moll′ , Jose′ Luis Vicedo. 2007. *Question Answering in Restricted Domains: An Overview*. Computational Linguistics, 33(1),

Dragomir Radev, WeiGuo Fan, Leila Kosseim. 2001. *The QUANTUM Question Answering System*. TREC.

Min-Yuh Day, Cheng-Wei Lee, Shih-Hung WU, Chormg-Shyong Ong, Wen-Lian Hsu. 2005. *An Integrated Knowledge-based and Machine Learning Approach for Chinese Question Classification*. Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering, Wuhan, China,:620~625.

Ulf Hermjakob. 2001. *Parsing and Question Classification for Question Answering*. Proceedings of the ACL Workshop on Open-Domain Question Answering, Toulouse,:19~25.

Dell Zhang, Wee Sun Lee. 2003. *Question classification using support vector machines*. Proceedings of the 26th Annual International ACM Conference on Research and Development in Information Retrieval(SIGIR), Toronto, Canada,26 ~ 32.

Valentin Jijkoun, Maarten de Rijke.2005. *Retrieving Answers from Frequently Asked Questions Pages on the Web*. CIKM'05, Bermen, Germany.

Noriko Tomuro. 2003. *Interrogative Reformulation Patterns and Acquisition of Question Paraphrases*. Proceeding of the Second International Workshop on Paraphrasing, :33~40.

Hyo-Jung Oh, Chung-Hee Lee, Hyeon-Jin Kim, Myung-Gil Jang. 2005. *Descriptive Question Answering in Encyclopedia*. Proceedings of the ACL Interactive Poster and Demonstration Sessions, pages 21–24, Ann Arbor.

Radu Soricut, Eric Brill. 2004, *Automatic Question Answering: Beyond the Factoid*. Proceedings of HLT-NAACL ,:57~64.

Jiwoon Jeon, W. Bruce Croft and Joon Ho Lee. 2005. *Finding Similar Questions in Large Question and Answer Archives*. CIKM'05, Bremen, Germany.

Sanda Harabagiu, Finley Lacatusu and Andrew Hickl. 2006 . *Answering Complex Questions with Random Walk Models*. SIGIR'06, Seattle, Washington, USA.pp220-227.

Ingrid Zukerman, Eric Horvitz. 2001. *Using Machine Learning Techniques to Interpret WH-questions*. ACL.

John Lafferty, Andrew McCallum, Fernando Pereira. 2001. *Conditional Random Fields: probabilistic Models for Segmenting and Labeling Sequence Data*. Proceedings of the Eighteenth International Conference on Machine Learning, p.282-289.

Nojun Kwak and Chong-Ho Choi. 2002. *Input feature selection for classification problems*. IEEE Trans on Neural Networks,,13(1):143-159

Context Inducing Nouns

Charlotte Price
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304 USA
lprice@parc.com

Valeria de Paiva
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304 USA
valeria.paiva@gmail.com

Tracy Holloway King
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304 USA
thking@parc.com

Abstract

It is important to identify complement-taking nouns in order to properly analyze the grammatical and implicative structure of the sentence. This paper examines the ways in which these nouns were identified and classified for addition to the BRIDGE natural language understanding system.

1 Introduction

One of the goals of computational linguistics is to draw inferences from a text: that is, for the system to be able to process a text, and then to conclude, based on the text, whether some other statement is true.[1] Clausal complements confound the process because, despite their surface similarity to adjuncts, they generate very different inferences.

In this paper we examine complement-taking nouns: how to identify them and how to incorporate them into an inferencing system. We first discuss what we mean by complement-taking nouns (section 2) and how to identify a list of such nouns (section 3). We then describe the question-answering system that uses the complement-taking nouns as part of its inferencing (section 4), how the nouns are added to the system (section 5), and how the coverage is tested (section 6). Finally, we discuss several avenues for future work (section 7), including automating the search process, identifying other context-inducing forms, and taking advantage of cross-linguistic data.

[1] We would like to thank the Natural Language Theory and Technology group at PARC, Dick Crouch, and the three reviewers for their input.

2 What is a complement-taking noun?

Identifying complement-taking nouns is somewhat involved. It is important to identify the clause, to ensure that the clause is indeed a complement and not an adjunct (e.g. a relative clause or a purpose infinitive), and to figure out what is licensing the complement, as it is not only nouns that license complements.

2.1 Verbal vs. nominal complements

A clause is a portion of a sentence that includes a predicate and its arguments. Clauses come in a variety of forms, a subset of which is shown in (1) for verbs taking complements. The italicized part is the complement, and the part in bold is what licenses it. The surface form of the clause can vary significantly depending on the licensing verb.

(1) a. Mary **knows** *that Bob is happy.*
b. John **wants** *(Mary) to leave right now.*
c. John **likes** *fixing his bike.*
d. John **let** *Mary fix his bike.*

For this paper, we touch briefly on nouns taking *to* clauses, as in (2b), but the main focus is on *that* clauses, as in (2a).

(2) a. the *fact* that Mary hopped
b. the *courage* to hop

Both types of complements pose problems in mining corpora for lexicon development. The *that* clauses can superficially resemble relative clauses, as in (3), and the *to* clauses can resemble purpose infinitives, as in (4).

(3) a. COMPLEMENT-TAKING NOUN: John liked the idea that Mary sang last evening.

Coling 2008: Proceedings of the workshop on Knowledge and Reasoning for Answering Questions, pages 9–16
Manchester, August 2008

b. RELATIVE CLAUSE: John liked the song that Mary sang last evening.

(4) a. COMPLEMENT-TAKING NOUN: John had a chance to sing that song.

b. PURPOSE INFINITIVE: John had a song book (in order) to sing that song.

As discussed in section 3, this superficial resemblance makes the automatic identification of complement-taking nouns very difficult: simple string-based searches would return large numbers of incorrect candidates which would have to be vetted before incorporating the new nouns into the system.

2.2 Contexts introduced by nominals

Complements and relative clause adjuncts allow very different inferences. Whereas the speaker's beliefs about adjuncts take on the truth value of the clause they are embedded in, the truth value of clausal complements is also affected by the licensing noun. Compare the sentences below. The italicized clause in (5) is a complement, while in (6) it is an adjunct.

(5) **The lie** *that Mary was ill* paralyzed Bob. $\Longrightarrow$ Mary was *not* ill.

(6) The **situation** *that she had gotten herself into* paralyzed Bob. $\Longrightarrow$ She had gotten herself into a **situation**.

To explain how this is possible, we introduce the notion of implicative contexts (Nairn et al., 2006), and claim that complement-taking nouns introduce a context for the complement, whereas no such context is created for the adjuncts. Perhaps the easiest way to think of a context is to imagine embedding the complement in an extra layer, with the layer adding information about how to adjust the truth-value of its contents.[2] This allows us to conclude in (5) that the speaker believes that *Mary* and *Bob* exist, as does the event of Bob's paralysis, but the event *Mary was ill* does not. These are refered to as the (un)instantiability of the components in the sentence. Contexts can be embedded within each other recursively, as in (7). Note that these semantic contexts often, but not always, correspond to syntactic embedding.

(7) Paul believes [that John's lie [that Mary worries [that fish can fly]] surprised us].

Contexts may have an implication signature (Nairn et al., 2006) attached to them, specifying, for example, that the clause is something that the speaker presupposes to be true or that the speaker believes the truth value of the clause should be reversed. The default for a context is to allow no implications to be drawn, as in (1b), where the speaker has not committed to whether or not Mary is leaving.

Below is a more detailed example showing how the context introduced by a noun changes the implications of the sentence, and how it would behave differently from a relative clause adjunct to a noun. Consider the pair of sentences in (8).

(8) a. The lie that Mary had won surprised John. $\Longrightarrow$ Mary did not win.

b. The bonus that Mary had won surprised John. $\Longrightarrow$ Mary won a bonus.

In (8), that John was surprised is in the speaker's top context, which is what the author commits to as truth. In (8a), *lie* is within the context of *surprised. Surprised* does not change the implications of elements within its context.[3] Therefore, *lie* gets a true value: that a lie was told is considered true. *That Mary won*, however, is within the context of *lie*, which reverses the polarity of implications within its scope or context. If *that Mary won* were only within the context of *surprised* instead of within *lie*, which would be the case if *lie* did not create a context, then *that Mary won* would fall within the context of *surprised.* The implication signature of *surprised* would determine the veridicality of the embedded clause instead of the signature of *lie*: this would incorrectly allow the conclusion that *Mary won*.

The content of the relative clause in (8b) is in the same context as *surprise* since no additional context is introduced by *bonus.* As such, we can conclude that Mary did win a bonus.

2.3 Complements introduced by *to*

The previous subsection focused on finite complements introduced by *that.* From the perspective

[2] In the semantic representations, the contexts are flattened, or projected, onto the leaf nodes of the parse tree, so that every leaf has access to information locally.

[3] We say *surprise* has the implication signature ++/--: elements within its context have a positive implication in a positive context and negative in a negative context. See (Nairn et al., 2006) for detailed discussion of possible implication signatures and how to propagate them through contexts.

of aiding inferencing in the BRIDGE system, the nouns that take *to* complements that are not deverbal nouns (see section 2.4 for discussion of deverbals) seem to fall into three main classes:[4] ability, bravery, and chance. Examples are shown in (9).

(9) a. John has the ability to sing.
b. John has the guts to sing out loud.
c. John's chance to sing came quickly.

These all have an implication signature that gives a (negative) implication only in a negative context, as in (10); in a positive context as in (9), no implication can be drawn.

(10) John didn't have the opportunity to sing.
$\Longrightarrow$ John didn't sing.

Note also that the implication only applies when the verb is *have*. Other light verbs, such as *take* in (11) change the implications.

(11) John took the opportunity to sing.
$\Longrightarrow$ John sang.

For this reason, these nouns are treated differently than those with *that* complements. They are marked in the grammar as taking a complement in the same way that *that* complements are (section 5), but the mechanism which attaches an implication signature takes the governing verb into account.

2.4 Deverbal nouns

A large number of complement-taking nouns are related to verbs that take complements. These nouns are analyzed differently than non-deverbal nouns. They are linked to their related verb and classified according to how the arguments of the noun and the sentence relate to the arguments for the verb (e.g. *-ee, -er*).[5] The BRIDGE system uses this linking to map these nouns to their verbal counterparts and to draw conclusions of implicativity as if they were verbs, as explained in (Gurevich et al., 2006). Consider (12) where the paraphrases using *fear* as a verb or a noun are clearly related.

(12) a. The fear that Mary was ill paralyzed Bob.
b. Bob feared that Mary was ill; this fear paralyzed Bob.

Deverbal nouns can take *that* complements or, as in (13), *to* complements. Most often, the context introduced by a deverbal noun does not add an implication signature, as in (11), which results in the answer UNKNOWN to the question *Was Mary ill?*.

(13) a. John's promise to go swimming surprised us.
b. John's persuasion of Mary to sing at the party surprised us.

Gerunds, being even more verb-like, are treated as verbs in our system and hence inherit the implicative properties from the corresponding verb.

(14) Knowing that Mary had sung upset John.
$\Longrightarrow$ Mary sang.

Gerunds and deverbal nouns are discussed in detail in (Gurevich et al., 2006) and are outside of the scope of this paper.

3 Finding complement-taking nouns

In order for the system to draw the inferences discussed above, the complement-taking nouns must first be identified and then classified and incorporated into the BRIDGE system (section 4). First, the gerunds are removed since these are mapped by the syntax into their verbal counterparts. Then the non-gerund deverbal nouns (section 2.4) are linked to their verbal counterpart so that they can be analyzed by the system as events. These two classes represent a significant number of the nouns that take *that* complements.

3.1 Syntactic classification

However, there are many complement-taking nouns that are not deverbal. To expand our lexicon of these nouns, we started with a seed set garnered from the Penn Treebank (Marcus et al., 1994), which uses distinctive tree structures for complement-taking nouns, and a small list of linguistically prominent nouns. For each of these lexical items, we extracted words in the same semantic class from WordNet. Classes include words like *fact*, which direct attention to the clausal complement, as in (15), and nouns expressing emotion, as in (16).

(15) It's a fact that Mary came.

(16) Bob's joy that Mary had returned reduced him to tears.

[4] The work described in this section was done by Lauri Karttunen and Karl Pichotta (Pichotta, 2008).

[5] NOMLEX (Macleod et al., 1998) is an excellent source of these deverbal nouns.

These semantic classes provided a starting point for discovering more of these nouns: the class of emotion nouns, for example, has more than a hundred hyponyms.

Identifying the class is not enough, as not all members take clausal complements. Compare *joy* in (16) and *warmheartedness* in (17) from the emotion class. The sentence containing *joy* is much more natural than that in (17).

(17) #Bob's warmheartedness that Mary had returned reduced him to tears.

From the candidate list, the deverbal nouns are added to the lexicon of deverbal noun mappings. The remaining list is checked word-by-word. To ease the process, test sentences that take a range of meanings are created for each class of nouns, as in (18).

(18) Bob's ___ that Mary visited her mother reduced him to tears.

If the noun does not fit the test sentences, a web search is done on *"X that"* to extract potential complement-bearing sentences. These are checked to eliminate sentences with adjuncts, or where some other feature licenses the clause, such as in (19) where the bold faced structure is licensing the italicized clause.

(19) a. John is **so** warmhearted *that he took her in without question.*

b. They had **such** a good friendship *that she could tell him anything.*

Using these methods, from a seed set of 13 nouns, ∼170 non-deverbal complement-taking nouns were identified, most in the emotion and feeling classes. The same techniques were then applied to the *state* and *information* classes. Once the Penn Treebank seeds were incorporated, the same process was applied to the complement-taking nouns from NOMLEX (Macleod et al., 1998).

3.2 Determining implications

As examples (8a) and (8b) showed, whether a word takes a complement is lexically determined; so is the type of implication signature introduced by the word. Compare the implications in (20).

(20) a. The **fact** that Mary had returned surprised John. $\Longrightarrow$ Mary had returned.

b. The **falsehood** that Mary had returned surprised John. $\Longrightarrow$ Mary had *not* returned.

c. The **possibility** that Mary had returned surprised John. ?$\Longrightarrow$ Mary had returned.

These nouns have different implication signatures: facts imply truth; lies imply falsehood; and possibilities do not allow truth or falsehood to be established. The default for complements is that no implications can be drawn, as in (20c), which in the BRIDGE system is expressed as the noun having no implication signature.[6]

Once identified and its implication signature determined, adding the complement-taking noun to the BRIDGE system and deriving the correct inferences is straightforward. This process is described in section 5.

4 The BRIDGE system

The BRIDGE system (Bobrow et al., 2007) includes a syntactic grammar, a semantics rule set (Crouch and King, 2006), an abstract knowledge representation (AKR) rule set, and an entailment and contradiction detection (ECD) system. The syntax, semantics, and AKR all depend on lexicons.

The BRIDGE grammar defines syntactic properties of words, such as predicate-argument structure, tense, number, and nominal specifiers. The grammar produces a packed representation of the sentence which allows ambiguity to be dealt with efficiently (Maxwell and Kaplan, 1991).

The parses are passed to the semantic rules which also work on packed structures (Crouch, 2005). The semantic layer looks up words in a Unified Lexicon (UL), connects surface arguments of verbs to their roles, and determines the context within which a word occurs in the sentence. Negation introduces a context, as do the complement-taking nouns discussed here (Bobrow et al., 2005).

The UL combines several sources of information (Crouch and King, 2005). Much of the information comes from the syntactic lexicon, VerbNet (Kipper et al., 2000), and WordNet (Fellbaum, 1998), but there are also handcoded entries that add semantically relevant information such as its implication signature. A sample UL entry is given in Figure 1.

The current number of complement-taking nouns in the system is shown in (21). Only a

[6] A context is still generated for these. Adjuncts, having no context of their own, inherit the implication signature of the clause containing them (section 2.2).

(cat(N), word(fact), subcat(NOUN-EXTRA),
concept(%1),
source(hand_annotated_data), source(xle),
xfr:concept_for(%1,fact),
xfr:lex_class(%1,impl_pp_nn),
xfr:wordnet_classes(%1,[])).

Figure 1: One entry for the word *fact* in the Unified Lexicon. NOUN-EXTRA states that this use of *fact* fits in structures such as *it is a fact that...* The WordNet meaning is found by looking up the concept for *fact* in the WordNet database. The implication signature of the word is impl_pp_nn or ++/--as seen in (22). Lastly, the sources for this information are noted.

fifth of the nouns have implication signatures. However, all of the nouns introduce contexts; the default implication for contexts is to allow neither *true* nor *false* to be concluded, as in (20c).

(21)

Complement-taking Nouns	
that complements	411
to complements	173
with implication signatures	107

The output of the semantics level is fed into the AKR. At this level, contexts are used to determine (un)instantiability based on the relationship between contexts.[7] An entity's (un)instantiability encodes whether it exists in some context. In (8a), for example, we can conclude that the speaker believes that *Mary* exists, but that the event *Mary won* is uninstantiated: the speaker believes it did not happen.

The final layer is the ECD, which uses the structures built by the AKR to reason about a given passage-query pair to determine whether or not the query is inferred by the passage, answering with YES, NO, UNKNOWN, or AMBIGUOUS. For more details, see (Bobrow et al., 2005).

5 Adding complement-taking nouns to the system

Adding complement-taking nouns to the BRIDGE system is straightforward. A syntactic entry is added indicating that the noun takes a complement. The syntactic classes are defined by templates, and the relevant template is called in the lexical entry for that word. For example, the template call @(NOUN-EXTRA %stem) is added to the entry for *fact*.

If there is an implication signature for the complement, this is added to the noun's entry in the file for hand-annotated data used to build the UL. The fifth line in Figure 1 is an example. The AKR and ECD rules that calculate the context and implications on verbs and deverbal nouns generalize to handle implications on complement-taking nouns and so do not need to be altered as new complement-taking nouns are found.

As described in section 3, deciding which nouns take complements is currently hand curated, as it is quite difficult to distinguish them entirely automatically.

6 Testing

To ensure that complement-taking nouns are working properly in the system, for each noun, a passage-query-correct answer triplet such as:

(22) PASSAGE: The fact that Mary had returned surprised John.
QUERY: Had Mary returned?
ANSWER: YES

is added to a testsuite. The testsuites are run and the results reported as part of the daily regression testing (Chatzichrisafis et al., 2007). Both naturally occurring and hand-crafted examples are used to ensure that the correct implications are being drawn. Natural examples test interactions between phenomena such as noun complementation and copular constructions, while hand-crafted examples allow isolation of the phenomenon and show that all cases are being tested (Cohen et al., 2008), e.g., that the correct entailments emerge under negation as well as in the positive case.

Our current testsuites contain about 180 hand-crafted examples. The number of natural examples is harder to count as they occur somewhat rarely in the mixed-phenomena testsuites. One of our natural example files, which is based on newswire extracts from the PASCAL Recognizing Textual Entailment Challenge (Dagan et al., 2005), shows an approximate breakdown of the uses of the word *that* is as shown in (23). This sample, which is somewhat biased towards verbal complements since it contains many examples that can be paraphrased as *said that*, nonetheless shows the relative scarcity of noun complements in the wild and underscores the importance of hand-crafted examples

[7] See (Bobrow et al., 2007; Bobrow et al., 2005) for other information contained in the AKR.

for testing purposes. It it is clear that these noun complements were being analyzed incorrectly before; what is unclear is how much of an impact the misanalysis would have caused. Perhaps some other domain would demonstrate a significantly higher presence of non-deverbal nouns that take complements and would be more significantly impacted by their misanalysis.

(23)

Uses of the word *that* in RTE 2007	
verbal complements	68
adjuncts	50
deverbal complements	14
noun complements	3
other [8]	19

7 Future work

The detection and incorporation of noun complements for use in the BRIDGE system can be expanded in several directions, such as automating the search process, identifying and classifying other parts of speech that take complements, and exploring transferability to other languages.

7.1 Automating the search

Testing whether a clause is an adjunct or a noun complement or is licensed by something else is currently done by hand. Automating the testing would allow many more nouns to be tested. However, this is non-trivial. As (8a) and (8b) demonstrated, the surface structure can appear very similar; it is only when we try to figure out the implications of the examples that the differences emerge.

The Penn Treebank (Marcus et al., 1994) was initially used to extract complement-taking nouns. As more tree and dependency banks, as well as lexical resources (Macleod et al., 1998), are available, further lexical items can be extracted in this way. However, such resources are costly to build and so are only slowly added to the available NLP resources.

Rather than trying to identify all potential noun complement clauses, a simpler approach would be to reduce the search space for the human judge. For example, some adjuncts (perhaps three quarters of them) could be eliminated from natural examples by using a part-of-speech tagger to identify occurrences where a conjugated verb immediately follows the word *that*, as in (24). These commonly identify adjuncts.

(24) The shark that *bit* the swimmer appears to have left.

By eliminating these adjuncts and by removing those sentences where it is known that the clause is a complement of the verb based on the syntactic classification of that verb (the syntactic lexicon contains ∼2500 verbs with various clausal complements), as in (25), the search space could be significantly reduced.

(25) The judge **announced** that the defendant was guilty.

7.2 Other parts of speech that introduce contexts

Verbs, adjectives, and adverbs can also license complements and hence contexts with implication signatures. Examples in (26) show different parts of speech that introduce contexts.[9]

(26) a. **Verb:** John **said** that Paul had arrived.

b. **Adjective:** It is **possible** that someone ate the last piece of cake.

c. **Adjective:** John was available *to see Mary*.

d. **Adverb:** John **falsely** reported that Mary saw Bill.

Many classes of verbs have already been identified and are incorporated into the system (Nairn et al., 2006): verbs relating to speech (e.g., *say*, *report*, etc.), implicative verbs such as *manage* and *fail* (Karttunen, 2007), and factive verbs (e.g. *agree*, *realize*, *consider*) (Vendler, 1967; Kiparsky and Kiparsky, 1971), to name a few. Many adjectives have also been added to the system, including ones taking *to* and *that* complements.[10] As with the complement-taking nouns, a significant part of the effort in incorporating the complement-taking adjectives into the system was identifying which adjectives license complements. The adverbs have not been explored in as much depth.

[8] This includes demonstrative uses, uses licensed by other parts of speech such as *so*, and clauses which are the subject of a sentence or the object of a prepositional phrase.

[9] From a syntactic perspective, the adverb *falsely* does not take a complement. However, it does introduce a context in the semantics and hence requires a lexical entry similar to those discussed for the complement-taking nouns.

[10] This work was largely done by Hannah Copperman during her internship at PARC.

7.3 Other languages

The fact that it has been productive to search for complement-taking nouns through synonyms and WordNet classes suggests that other languages could benefit from the work done in English. It would be interesting to see to what extent the implicative signatures from one language carry over into another, and to what extent they differ. Strong similarities could, for example, suggest some common mechanism at work in these nouns that we have been unable to identify by studying only one language. Searching in other languages could also potentially turn up classes or candidates that were missed in English.[11]

8 Conclusions

It is important to identify complement-taking nouns in order to properly analyze the grammatical and implicative structure of the sentence. Here we described a bootstrapping approach whereby annotated corpora and existing lexical resources were used to identify complement-taking nouns. WordNet was used to find semantically similar nouns. These were then tested in closed examples and in Web searches in order to determine whether they licensed complements and what the implicative signature of the complement was. Although identifying the complete set of these nouns is non-trivial, the context mechanism for dealing with implicatives makes adding them to the BRIDGE system to derive the correct implications straightforward.

9 Appendix: Complement-taking nouns

This appendix contains sample complement-taking nouns and their classification in the BRIDGE system.

9.1 Noun that take *to* clauses

Ability nouns (impl_nn with verb *have*): ability, choice, energy, flexibility, freedom, heart, means, way, wherewithal

Asset nouns (impl_nn with verb *have*): money, option, time

Bravery nouns (impl_nn with verb *have*): audacity, ball, cajones, cheek, chutzpah, cojones, courage, decency, foresight, gall, gumption, gut, impudence, nerve, strength, temerity

Chance nouns (impl_nn with verb *have*): chance, occasion, opportunity

Effort nouns (impl_nn with verb *have*): initiative, liberty, trouble

Other nouns (no implicativity or not yet classified): accord, action, agreement, aim, ambition, appetite, application, appointment, approval, attempt, attitude, audition, authority, authorization, battle, bid, blessing, campaign, capacity, clearance, commission, commitment, concession, confidence, consent, consideration, conspiracy, contract, cost, decision, demand, desire, determination, directive, drive, duty, eagerness, effort, evidence, expectation, failure, fear, fight, figure, franchise, help, honor, hunger, hurry, idea, impertinence, inability, incentive, inclination, indication, information, intent, intention, invitation, itch, job, journey, justification, keenness, legislation, license, luck, mandate, moment, motion, motive, move, movement, need, note, notice, notification, notion, obligation, offer, order, pact, pattern, permission, plan, pledge, ploy, police, position, potential, power, pressure, principle, process, program, promise, propensity, proposal, proposition, provision, push, readiness, reason, recommendation, refusal, reluctance, reminder, removal, request, requirement, responsibility, right, rush, scheme, scramble, sense, sentiment, shame, sign, signal, stake, stampede, strategy, study, support, task, temptation, tendency, threat, understanding, undertaking, unwillingness, urge, venture, vote, willingness, wish, word, work

9.2 Nouns that take *that* clauses

Nouns with impl_pp_nn: abomination, angriness, angst, animosity, anxiousness, apprehensiveness, ardor, awe, bereavement, bitterness, case, choler, consequence, consternation, covetousness, disconcertion, disconcertment, disquiet, disquietude, ecstasy, edginess, enmity, enviousness, event, fact, fearfulness, felicity, fright, frustration, fury, gall, gloom, gloominess, grudge, happiness, hesitancy, hostility, huffiness, huffishness, inquietude, insecurity, ire, jealousy, jitteriness, joy, joyousness, jubilance, jumpiness, lovingness, poignance, poignancy, premonition, presentiment, problem, qualm, rancor, rapture, sadness, shyness, situa-

[11] Thanks to Martin Forst (p.c.) for suggesting this direction.

tion, somberness, sorrow, sorrowfulness, suspense, terror, trepidation, truth, uneasiness, unhappiness, wrath

Nouns with fact_p: absurdity, accident, hypocrisy, idiocy, irony, miracle

Nouns with impl_pn_np: falsehood, lie

Other nouns (no implicativity or not yet classified): avowal, axiom, conjecture, conviction, critique, effort, fear, feeling, hunch, hysteria, idea, impudence, inability, incentive, likelihood, news, notion, opinion, optimism, option, outrage, pact, ploy, point, police, possibility, potential, power, precedent, premise, principle, problem, prospect, proviso, reluctance, responsibility, right, rumor, scramble, sentiment, showing, sign, skepticism, stake, stand, story, strategy, tendency, unwillingness, viewpoint, vision, willingness, word

References

Bobrow, Daniel G., Cleo Condoravdi, Richard Crouch, Ron Kaplan, Lauri Karttunen, Tracy Holloway King, Valeria de Paiva, and Annie Zaenen. 2005. A basic logic for textual inference. In *Proceedings of the AAAI Workshop on Inference for Textual Question Answering*.

Bobrow, Daniel G., Bob Cheslow, Cleo Condoravdi, Lauri Karttunen, Tracy Holloway King, Rowan Nairn, Valeria de Paiva, Charlotte Price, and Annie Zaenen. 2007. PARC's Bridge and question answering system. In *Grammar Engineering Across Frameworks*, pages 46–66. CSLI Publications.

Chatzichrisafis, Nikos, Dick Crouch, Tracy Holloway King, Rowan Nairn, Manny Rayner, and Marianne Santaholma. 2007. Regression testing for grammar-based systems. In *Grammar Engineering Across Frameworks*, pages 28–143. CSLI Publications.

Cohen, K. Bretonnel, William A. Baumgartner Jr., and Lawrence Hunter. 2008. Software testing and the naturally occurring data assumption in natural language processing. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 23–30. Association for Computational Linguistics.

Crouch, Dick and Tracy Holloway King. 2005. Unifying lexical resources. In *Proceedings of the Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes*.

Crouch, Dick and Tracy Holloway King. 2006. Semantics via f-structure rewriting. In *LFG06 Proceedings*. CSLI Publications.

Crouch, Dick. 2005. Packed rewriting for mapping semantics to KR. In *Proceedings of the International Workshop on Computational Semantics*.

Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognizing textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*, Southampton, U.K.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Gurevich, Olga, Richard Crouch, Tracy Holloway King, and Valeria de Paiva. 2006. Deverbal nouns in knowledge representation. In *Proceedings of the 19th International Florida AI Research Society Conference (FLAIRS '06)*, pages 670–675.

Karttunen, Lauri. 2007. Word play. *Computational Linguistics*, 33:443–467.

Kiparsky, Paul and Carol Kiparsky. 1971. Fact. In Steinberg, D. and L. Jakobovits, editors, *Semantics. An Inderdisciplinary Reader*, pages 345–369. Cambridge University Press.

Kipper, Karin, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *AAAI-2000 17th National Conference on Artificial Intelligence*.

Macleod, Catherine, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. NOMLEX: A lexicon of nominalizations. In *EURALEX'98*.

Marcus, Mitchell, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies adn Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn treebank: Annotative predicate argument structure. In *ARPA Human Language Technology Workshop*.

Maxwell, John and Ron Kaplan. 1991. A method for disjunctive constraint satisfaction. *Current Issues in Parsing Technologies*.

Nairn, Rowan, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Inference in Computational Semantics (ICoS-5)*.

Pichotta, Karl. 2008. Processing paraphrases and phrasal implicatives in the Bridge question-answering system. Stanford University, Symbolic Systems undergraduate honors thesis.

Vendler, Zeno. 1967. *Linguistics and Philosophy*. Cornell University Press.

Know-Why Extraction from Textual Data for Supporting What Question

Chaveevan Pechsiri
Dept. of Information Technology,
DhurakijPundit University,
Bangkok, Thailand
itdpu@hotmail.com

Phunthara Sroison
Dept. of Information Technology,
DhurakijPundit University,
Bangkok, Thailand
phunthara@it.dpu.ac.th

U. Janviriyasopak
Eastern Industry Co.ltd.
Bangkok, Thailand
uraiwanjan@hotmail.com

Abstract

This research aims to automatically extract Know-Why from documents on the website to contribute knowledge sources to support the question-answering system, especially What-Question, for disease treatment. This paper is concerned about extracting Know-Why based on multiple EDUs (Elementary Discourse Units). There are two problems in extracting Know-Why: an identification problem and an effect boundary determination problem. We propose using Naïve Bayes with three verb features, a causative-verb-phrase concept set, a supporting causative verb set, and the effect-verb-phrase concept set. The Know-Why extraction results show the success rate of 85.5% precision and 79.8% recall.

1 Introduction

Automatically Know -Why extraction is essential for providing the rational knowledge source, to the society through question answering system, especially in herbal medicines when assisting the locals to understand more about herbs. According to Jana Trnkova and Wolfgang Theilmann (2004) Know-Why is the knowing of the reason of why something is the way it is. Therefore, Know-Why has to involve the causal relation which is "an irreflexive, transitive and asymmetrical" relation that contains the properties of "productivity (effect is 'produced' by the cause) and locality (it obeys the markov condition, for model A→ B → C, if there is no B, then A does not cause C)"(Lemeire J. et al. (2004)). Wolff P. (2007) stated that the causal relation can be decomposed into 2 major approaches, the dependency model and the physicalist models. The dependency model can be represented by using statistical dependency model whereas in recent physicalist models are based on the concepts of force dynamic models consisting of 2 force entities in certain events; the agonist and the antagonist (Talmy, 2000). Later, the agonist form (Wolff P., 2007) can be viewed as the 'effect' and the antagonist as the 'cause'. According to Talmy (2000), if there is a situation where the antagonist is stronger, which can be expressed as 'event X happens because of event Y'(Y contains the antagonist.), it is a form of causation. Moreover, the causal relation can pivot on the distinction between causality and causation (Lehmann J. et al, 2004) whereas causality is 'a law-like relation between cause events and effect events' and causation is 'the actual causal relation that holds between individual events'. For example:

"Because a bird sings a song at a window, The rock is thrown at the window."

Causality: "An object vibrates. An object moves."

Causation: "A bird sings. The rock is thrown"

This research focuses only on 'causal relation' to provide both 'causality' for extracting Know-Why from the herbal medicine domain and 'causation' for answering What-question, since what questions contain ambiguities (Girju R. and Moldovan D., 2002) for example:

Know-Why: "*ใบกระเพราใช้เป็นยาขับลม แก้คลื่นไส้ แก้ปวดท้อง /A basil leaf is used as a medicine releasing gas. [The leaf] stops nausea. [The leaf] stops paining the abdomen.*" (where the [..] symbol means ellipsis.)

Coling 2008: Proceedings of the workshop on Knowledge and Reasoning for Answering Questions, pages 17–24
Manchester, August 2008

Know-Why concept: "*A herb organ is used as being a carminative drug. [The organ] is anti nausea, [The organ] is anti stomachache*."
Question: "*ใช้สมุนไพรอะไรแก้คลื่นไส้/What herb is used for stopping nausea*?" From this example, '*A basil leaf is used as a medicine releasing gas'* is the causation and the concept is the causality.

There are various forms of causal-relation expression such as in the form of intra-NP, inter-NP, and inter-sentence (Chang and Choi,2004). According to our research, we separated this relation into 2 main forms based on the elementary discourse unit (EDU) as defined by (Carlson et al., 2003) as a simple sentence or clause. We defined the intra-causal EDU as an expression within one simple EDU being equivalent to either the intra-NP form or the inter-NP form (Chang and Choi,2004). The inter-causal EDU is defined as an expression within more than one simple EDU which is equivalent to the inter-sentences of Chang and Choi (2004). However, this paper works on only the inter-causal EDU extraction because some cause-effect relation from the herbal web sites are expressed in the form of the EDU containing an EDU-like name entity with the causative action followed by some effect EDUs.

Several techniques (Marcu and Echihabi,2002; Torisawa 2003; Inui and et al.,2004; Pechsiri and Kawtrakul, 2007) have been used to extract cause-effect knowledge varying from two adjacent sentences to multiple sentences. Our work aimed at mining and extracting Know-Why from Thai documents of herbal medicines. Thai has several specific characteristics, such as the existence of sentence-like name entity, zero anaphora or the implicit noun phrase. All of these characteristics are involved in the two main problems of Know-Why extraction: the first problem is how to identify the interesting causality events expressed by an EDU- like name entity from documents, and the second one is how to identify the effect boundary, where The problem of implicit delimiter of the boundary is involved. From all of these problems, we needed to develop a framework which combineed Language Processing and the machine learning technique as Naïve Bayes to learn features of three verb sets, a causative concept verb set, a supporting causative verb set, and an effect concept verb set, for solving those problems.

In conclusion, unlike other methods (Marcu and Echihabi ,2002; Torisawa 2003; Inui and et al.,2004) where the emphasis is based on two adjacent sentences, this paper is based on multiple EDU extraction. Our research was separated into 5 sections. In section 2, related work was summarized. Problems in causality mining from Thai documents will be described in section 3 and in section 4 our framework for causality extraction was explained. In section 5, we evaluated and concluded our proposed model.

2 Related Work

Several strategies such as those done by Marcu and Echihabi ,2002, Torisawa(2003), Inui and et al.(2004), and Pechsiri and Kawtrakul (2007) have been proposed to extract and discover knowledge from the textual data.

Marcu and Echihabi (2002) presented the unsupervised approach to recognize the discourse relations by using word pair probabilities between two adjacent sentences for classifying the rhetorical relations, such as Contrast, Cause-Explanation, Condition, and Elaboration, between two adjacent sentences by using Naïve Bayes classifier to the BLIPP corpus (Charniak, 2000). They determined the word pairs in the cartesian product from the sentence pairs connected with or without discourse marker or connective marker , i.e. 'because' 'but' 'then', to classify the causal relation from other rhetorical relations. The result showed an accuracy of 75% of inter-sentence causality extraction from the corpus size of more than a million sentences for learning whereas our corpus size is 3000 sentences for learning. Therefore, our approach is the supervised approach with the statistical method because our corpus size is small.

Inui's work (Inui and et al.,2004) proposed a method of extraction and classification of causal knowledge. The method of extraction was accomplished under two adjacent sentences by using explicit connective markers; e.g. "because" "since" "if..then" "as the result" etc.. SVM was used for the classification process in (Inui and et al.,2004). Four types of causal relations are studied, including the following: cause, precondition, mean, effect relations. Inui's work's precision is high: 90% but the recall is low: 30%, because of unresolved anaphora. However, in our work, we extract multiple EDUs with some implicit discourse markers.

Torisawa(2003)' s work in extracting the verb phrase pair from the news corpus worked on the assumption that if two events share a common participant (is specified by a noun) then the two events are likely to have a logical relation as causal relation. For example "A man drank

liquor and was intoxicated by the liquor."(a common participant is 'liquor'). However, this assumption can not be applied in our research because most of our causality expression does not share a common participant; e. g. "*ขิงใช้เป็นยาระบาย แก้ท้องผูก/Ginger is used as being laxative medicine. [The ginger] stops constipation.*

Pechsiri and Kawtrakul (2007), proposed verb-pair rules learned by two different machine learning techniques (NB and SVM) to extract causality with multiple EDUs of a causative unit and multiple EDUs of an effect unit with the problems of the discourse marker ambiguity and the implicit discourse marker. This verb-pair rule has been represented by the following equation (1) (Pechsiri and Kawtrakul, 2007) where V_c is the causative verb concept set, V_e is the effect verb concept set , C is the Boolean variables of causality and non-causality, and a causative verb concept (v_c , where $v_c \in V_c$) and an effect verb concept (v_e , where $v_e \in V_e$) are referred to WordNet (http://wordnet.princeton.edu/) and the predefined plant disease information from Department of Agriculture (http://www.doa.go.th/).

CausalityFunction: $V_c \wedge V_e \rightarrow C$ (1)

They also proposed using V_c and V_e to solve the boundary of the causative unit and using the Centering theory along with V_e to solve the boundary of the effect unit. The outcomes of their research were the verb-pair rule, V_c, V_e, and the multiple EDUs of causality (extracted from textual data) was at their highest precision of 89% and their highest recall of 76%. The correctness of the causality-boundary determination is 88% on average. However, our causative unit consisted of only one EDU containing an EDU-like name entity as a cause, and this EDU was followed by several effect EDUs.

In our current work, we aimed at extracting the Know-Why in Natural Language description instead of visualizing only associations of concepts, by applying both language processing and learning technique by Naïve Bayes to identify the causality expression.

3 Problem of Know-Why Extraction

To extract the cause-effect expressions, there are two main problems that must be solved. The first problem is how to identify interesting cause-effect relations from the documents. The second problem is how to determine the effect boundary.

There is also the problem of implicit noun phrase.

3.1 Causality Identification

The problem involves the word level and the sentence level. For the word level, the medicinal name entity may express in the form of a sentence like name entity or an EDU- like name entity which explains the medicinal action as the causative action of medicine, and medical characteristic. The problem of this level is how to identify the causative name entity. For example:

a) "*ใบกระเพรา/**A basil leaf** ใช้เป็น/**is used as** ยา/**medicine** ขับ/**releases** ลม/**gas***"

where 'a medicine releases gas' is an EDU-like name entity with the causative action, 'release'.

b) "*แก่นกระแจะ/**Nicolson stem** ใช้ทำ/**is used for making** ยา/**medicine** ดอง/**soaks in** เหล้า/**liquor***"

where 'a medicine soaks in liquor' is an EDU-like name entity with the characteristic of medicine being preserved in the alcohol.

The above examples, a) and b), contain an EDU-like name entity which is a cause in a) and a non cause in b).

For the sentence level, the EDU containing an EDU-like name entity with the causative action may be followed by an effect EDU(s) to form the cause-effect or causality relation between the EDU like name entity and that following EDU(s). For example:

Causality

EDU1 "*ตะไคร้หอม/**Lemon grass** ใช้เป็น/**is used as** ยา/**medicine** บีบ/**contracts** ,มดลูก/**a uterus***" (where 'a medicine contracts a uterus.' is the EDU-like name entity with concept of 'the medicine causes uterus to contract'.)

EDU2 "*[**The plant**] ขับ/discharges ประจำเดือน/**period.***" (=The plant discharges period.)

Non causality

EDU1 "*ใบกระเพรา/**A basil leaf** ใช้เป็น/**is used as** ยา/**medicine** ขับ/**releases** ลม/**gas**.*" (where 'a medicine releases gas' is the causative EDU-like name entity.)

EDU2 "*[the basil leaf]*รักษา/***relieves*** แผล/***ulcer*** ใน/***in***กระเพาะอาหาร/***stomach***."(= [The basil relives ulcer in a stomach.)

Where in this example, EDU 1 is the cause and EDU2 is the effect

3.2 Effect Boundary Determination

There are two problems of an implicit effect boundary cue and the effect EDU containing interrupts.

3.2.1 *Implicit Effect Boundary Cue*

Some cause-effect relations from the herbal web sites are expressed in the form of the EDU containing an EDU like name entity with the causative action followed by some effect EDUs without any cue of ending effect boundary, e.g. "และ/ and". For example:

EDU1 "ใบกระเพรา***A basil leaf*** ใช้เป็น/***is used as*** ยา/ ***medicine*** ขับ/***releases*** ลม/***gas***" (=A basil leaf is used as a medicine releasing gas.)

EDU2 "*[**The basil leaf**]* แก้/***stops*** คลื่นไส้/ ***nauseate.***" (=The basil leaf stop being nausea.)

EDU3 "*[**And the leaf**]* แก้/***stops*** ปวด/***pain*** ท้อง/ ***abdomen***." (= [And the leaf] stops paining abdomen.)

Where in this example, EDU 1 is the cause and EDU 2 & EDU3 are the effects. EDU 2 and EDU3 help us to determine the boundary.

3.2.2 *Effect EDU Containing Interrupts*

There are some effect EDUs containing interrupts as shown in the following example:

EDU1 "หอมแดง/***A red onion*** ใช้เป็น/***is used as*** ยา/ ***medicine*** ถ่าย ***/be laxztive***" (=A red onion is used as a laxative medicine.)

EDU2 "*[And the red onion]* แก้/***stops*** ท้องผูก/ ***being constipation***" (= [And the red onion] stops being constipation.)

EDU3 "*[**The red onion**]* ขับ/***discharges*** ปัสสาวะ ***/urine.***" (= [The red onion] discharges urine.)

EDU4 "*[**The red onion** makes a patient]* เจริญ อาหาร/***be appetite***." (= [The red onion] makes a patient] be appetite.)

Where the EDU-like name entity in EDU1 is a cause with EDU2 and EDU4 as its effects. The EDU3 is an interrupt. Although EDU3 is the effect of red onions, but EDU 3 is not the effect of laxatives.

4 A Framework for Know-Why Extraction

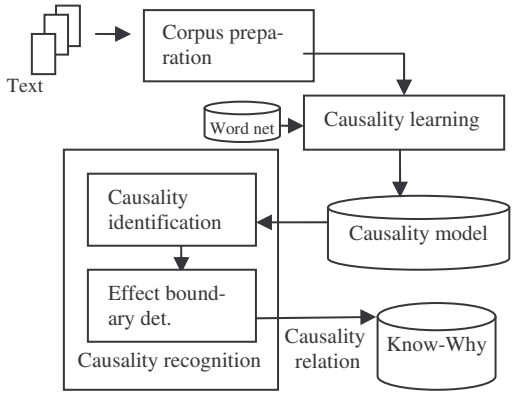

Figure 1. System Overview

There are three steps in our framework. First is the corpus preparation step followed by causality learning, and causality recognition steps (as shown in figure 1).

4.1 Corpus Preparation

There are two steps of pre-annotation and Causality annotation.

4.1.1 *Pre-annotation*

This step is the preparation of the corpus in the form of EDU from the text. The step involves using Thai word segmentation tools to solve a boundary of a Thai word and tagging its part of speech (Sudprasert and Kawtrakul, 2003). This process includes Name entity (Chanlekha and Kawtrakul, 2004), and word-formation recognition (Pengphom, et al 2002) to solve the boundary of Thai Name entity and Noun phrase.

After the word segmentation is achieved, EDU segmentation is dealt with. According to Charoensuk et al. (2005), this process segments plain text into units of EDUs by using the rule based and the machine learning technique of C4.5 (Mitchell T.M., 1997). These generated EDUs will be kept as an EDU corpus. This corpus will contain 4500 EDUs and will be separated into 2 parts, one part is 3500 EDUs for causality learning and the other part of 1000 EDUs for causality recognition and extraction.

4.1.2 *Causality Annotation*

Due to the problems in the causality identification, verbs from three EDUs (with one EDU as an EDU-like name entity) in the EDU corpus are used in this process to learn for extracting causality. Word ambiguity will be solved through the

finding of word concepts from Wordnet. Since Thai Wordnet does not exist, we need to translate from Thai to English, using Lexitron (the Thai-English dictionary)(http://lexitron.nectec.or.th/), before using Wordnet(http://wordnet.princeton.edu/obtain). In this process, we manually annotate the causality EDUs by annotating the EDU containing the causative EDU-like name entity as the causative EDU. We annotate a verb phrase in the causative EDU-like name entity to be a causative-verb-phrase concept (referred to Wordnet). The verb from EDU which contains the causative EDU-like name entity is annotated with a concept and we call this verb as 'supporting causative verb'. We also annotate the effect-verb-phrase concept(referred to Wordnet and http://www.ars-grin.gov/duke/ethnobot.html) from effect EDUs following the EDU containing the causative EDU-like name, as shown in Figure 2)

4.2 Causality Learning

The aim of this step was to learn cause-effect relation between causative events and effect events from annotating an EDU corpus.

4.2.1 *Feature Extraction*

All annotated verb features from the previous step are extracted into database table (in Table 1) including surface forms of verb features along with their concepts used for probability determination in the next step.

```
<C id=1>
<EDU type =cause>
  <NP1 concept=a herb organ>ใบโหระพา A basil leaf</NP1>
  <VS concept=use#1>ใช้เป็น is used as</VS>
   <EDU-Like-NE >
     <NP2 concept=drug>ยา medicine</NP2>
     <CVC concept= be carminative/ eliminate gas from a body>
       ขับ releases ลม/ gas
     </CVC>
   </EDU-Like-NE>
</EDU>
</C>
<R id=1>
<EDU type=effect>
   <EVC concept= stop nausea/ be anti nausea>แก้ stops คลื่นไส้/ nauseate.
   </EVC>
</EDU>
<EDU type=effect>
    <EVC concept=stop paining an abdomen/ relieve abdominal pain>แก้/
      stops ปวด/pain ท้อง/ abdomen
    </EVC>
</EDU>
</R>

EDU= EDU, EDU-Like-NE= EDU-like name entity tag,
C=cause tag, R=result or effect tag, VS= supporting verb tag ,
CVC=causative verb concept tag, EVC=effect verb concept tag
NP1 NP2= noun phrase tag
```

Figure 2. Causality Annotation Tag

NP1	NP1 Concept	Vs	Vs Concept	VPc	VPc concept	VPe	VPe Concept	Class
กระแจะ/ Naringi crenulata	herb	ใช้เป็น	use as	ดับพิษ/ cure poison	be-antipyretic	แก้ปวดเมื่อย	relieve muscle pain	n
บัวบก/ Asiatic Pennyworth	herb leaf	ใช้เป็น	use as	ทาภายนอก/ apply externally	apply topically	รักษาแผล	heal wound	y
หอมแดง/ red onion	herb	เป็น	is	ถ่าย/ excrete	be-lexative	แก้ท้องผูก	stop being constipation	y
หอมแดง/ red onion	herb	เป็น	is	ถ่าย/ excrete	be-lexative	ขับปัสสาวะ	discharge urine	n
หอมแดง/ red onion	herb	เป็น	is	ถ่าย/ excrete	be-lexative	เจริญอาหาร	be appetite	y
ขมิ้นชัน/ curcumin	herb	เป็น	is	ฆ่าเชื้อ/ antiseptic	be-antiseptic	รักษาโรคผิวหนัง	cure skin disease	y
มะแว้ง/ Soianum indicum Linn	herb	ทำเป็น	make as	ลดน้ำตาลในเลือด/ reduce blood sugar	balance blood sugar level	แก้ไอ	stop coughing	n
กระเพรา / Basil	herb leaf	ใช้เป็น	use as	ขับลม/ release gas	be carminative	แก้คลื่นไส้	relieve nausea	y
กระเพรา / Basil	herb leaf	ใช้เป็น	use as	ขับลม/ release gas	be carminative	แก้ปวดท้อง	stop paining an abdomen	y
ขิง / ginger	herb	ใช้เป็น	use as	ขับลม/ release gas	be carminative	แก้คลื่นไส้	relieve nausea	y
มะกูด / bergamot leaf	herb leaf	ใช้เป็น	use as	ขับลม/ release gas	be carminative	แก้คลื่นไส้	relieve nausea	y
…	…	…	…	…	…	…	…	…

Table 1. The extracted features from the annotated corpus

Vs concept	causality	non causality
ใช้+เป็น/use as	0.27619	0.290323
Be	0.561905	0.612903
ทำ+เป็น/make as	0.009524	0.032258
ใช้+ทำ+เป็น/use for making as	0.066667	0.053763
…	…	…
VPc concept	causality	non causality
‘ขับ+ลม/release-gas’	0.371901	0.192661
‘แก้+ไอ/anti coughing’	0.024793	0.045872
‘ทา/apply’	0.140496	0.009174
‘ขม/be-bitter’	0.041322	0.009174
‘ขับ+ปัสสวะ/discharge-urine’	0.057851	0.06422
‘ขับ+เสมหะ/be expectorant’	0.041322	0.06422
‘บีบ+มดลูก/contract uterus/oxytocic’	0.041322	0.027523
‘รักษา+เบาหวาน/be antidiabetic’	0.008264	0.027523
…	…	…
VPe concept	causality	non causality
‘แก้+ปวด+ท้อง/stop-stomachach/relieve abdominal pain’	0.035714	0.007813
‘แก้+คลื่นไส้/stop-naucea/be anti nausea’	0.035714	0.007813
‘แก้+ท้องอืดท้องเฟ้อ/stop-flatulence/relieve indigestion’	0.15	0.007813
‘แก้+ลมพิษ/stop-rash/ be antiurticaria’	0.035714	0.023438
‘ลดไข้/reduce-fever’	0.021429	0.039063
‘ขับ+รก/eliminate-placenta’	0.007143	0.054688
‘เริญ+อาหาร/increase appetite’	0.092857	0.007813
‘ขับ+เหงื่อ/release-sweat/be diaphoretic’	0.007143	0.070313

Table 2. Show probability of V_s concept, VP_c concept and VP_e concept

4.2.2 *Probability Determination*

After we had obtained the extracted verb features, we then determined the probability of causal and non causal from the occurrences of the cartesian products of three verb feature concepts , shown in Table2, by using Weka which is a software tool for machine learning (http://www.cs. waikato.ac.nz/ml/weka/).

4.3 Causality Recognition and Extraction

The objective of this step was to recognize and extract the cause-effect relation from the testing EDU corpus. In order to start the causality recognition process, Naïve Bayes Classifer shown in equation (2) is applied with the feature probabilities in Table 2, where EDUs class is determined by class1 (causality EDUs) and class0 (non causality EDUs).

$$EDUclass = \arg\max_{class \in Class} P(class | v_s, vp_c, vp_e) = \arg\max_{class \in Class} P(v_s | class) P(vp_c | class) P(vp_e | class) P(class) \quad (2)$$

$v_s \in V_s$ where V_s is a Supporting Causative Verb concept set

$vp_c \in VP_c$ where VP_c is a Causative VerbPhrase concept set

$vp_e \in VP_e$ where VP_e is a Effect VerbPhrase concept set

Therefore, Causality Recognition can be separated into 2 steps: causality identification and effect boundary determination.

4.3.1 *Causality Identification*

This step was to determine the interesting locations that are cause-effect relations by searching any EDU which consists of a verb matching to a verb in the supporting causative concept set, V_s, and an EDU-like name entity containing a causative-verb-phrase concept as vp_c (where $vp_c \in VP_c$).

4.3.2 *Effect Boundary Determination*

The effect EDU and the effect boundary were determined at the same time by checking all sequence EDUs right after the EDU containing vp_c in the EDU-like name entity. If a verb phrase from the sequence of checked EDUs is not in VP_e, the possible effect boundary is end. After the possible boundary is determined, v_{s_inEDU1}, vp_{c_inEDU1} and $vp_{e\_inEDU2}..vp_{e\_inEDUn}$ (where n>2) will be used to determine the causality class from the Naïve Bayes Classifier equation (2) as shown in Figure 3. The actual effect boundary is determined from the last class1 in the sequence of EDU_2.. EDU_n.

Furthermore, where the implicit noun phrase occurs as the subject of the current EDU, this has to be solved in this step by using the heuristic rule which is that the noun phrase as a subject of the previous EDU will be the subject of the current EDU.

Assume that each EDU is represented by (np vp)
L is a list of EDU
VP_C is a causative-verb-phrase concept set, VP_E /VPe is a effect-verb-phrase concept set
V_S is a supporting causative verb concept set
CAUSALITY_EXTRACTION (L, V_C, V_E, Vs)

```
1    i ← 1, R← ∅
2    while i ≤ length[L] do
3    begin_while1
4       CA ← ∅, EC ← ∅
5       if (vp_i ∈ V_S) ∧ (vp_i_in_NE ∈ VP_C) then
6        begin_if
           CA ← CA ∪ {i}, i ← i + 1   /*CA is causative EDU
7          while (vp_i ∈ VP_E) do
8           begin_while2
9       res ←
          arg max_{c∈(yes,no)} P(v_s |c)P(vp_c |c)P(vp_e |c)P(c)
10           if res=yes
11            EC ← EC ∪ {i},   /*EC is effect EDU
12             i ← i + 1
13            end_while2
14           endif
15      if res = yes ∧ CA <>∅ then
16           R = R ∪ { (CA,EC) }
17   end_while1
18    return R
```

Figure3. Show Causality Extraction algorithm for the EDU containing the causative EDU-like name entity, and followed by multiple effect EDUs .

5 Evaluation and Conclusion

The Thai corpora used to evaluate the proposed causality extraction algorithm consist of about 1,000 EDUs collected from several herbal web sites. The evaluation of the causality extraction performance of this research methodology is expressed in terms of the precision and the recall as shown below, where R is the causality relation:

$$Precision = \frac{\#of\ samples\ correctly\ extracted\ as\ R}{\#of\ all\ samples\ output\ as\ being\ R} \quad (3)$$

$$Recall = \frac{\#of\ samples\ correctly\ extracted\ as\ R}{\#of\ all\ samples\ holding\ the\ target\ relation\ R} \quad (4)$$

The results of precision and recall are evaluated by three expert judgments with max win voting. The precision of the extracted causality 85.5% with 79.8% recall. The correctness of our effect boundary determination by these expert judgments is 86%. These research results can be increased if we use a larger corpus. However, our methodology will be very beneficial for contribute the causality knowledge for supporting What-question with the concept of causal relation from a web page by inference method of backward chaining, for example:
Extracted causality: "*ใบโหระพาใช้เป็นยาขับลม แก้คลื่นไส้ แก้ปวดท้อง /A basil leaf is used for a gas released medicine. [The leaf] stops nausea. [The leaf] stop stomachache*." …….

The above extracted causality can be represented by the following predication.

a) ∀x be_herb(x) ^ be_herb_medicine(y) ^ be_carminative (y) ^ use_as(x,y)→ stop(x, z) ^ be_nausea(z)

b) ∀x be_herb(x) ^ be_herb_medicine(y) ^ be_carminative (y) ^use_as(x,y)→ stop(x, z) ^ be_abdominal pain(z)
Where x ∈ X,{'ใบกระเพรา/basil leaf' 'ขิง/ginger' 'พริกไทย/black pepper' 'ใบมะกรูด/bergamot leaf'..}, and X is the extracted NP1 set from EDUs containing the causative EDU-like name entities and being followed by the effect EDUs , e.g. (stop(x, z) ^ be_nausea(z)), (stop(x, z) ^ be_stomach ache(z)).

Question: "ใช้/***use*** สมุนไพร/***herb*** อะไร/***what*** แก้/***stop*** คลื่นไส้/***nausea*** *(What kind of herb is used for stop nausea*?)
The backward chaining from the above question and the extracted causality in a) is shown in the following
stop(x, z) ^ be_nausea(z) →be_herb(x) ^ be_herb_medicine(y) ^ be_carminative (y) ^ use_as(x,y)

where x is 'ใบกระเพรา/basil leaf', 'ขิง/ginger', 'พริกไทย/black pepper', or 'ใบมะกรูด/bergamot leaf'

References

Carlson L., Marcu D., and Okurowski M. E. 2003. Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In Current Directions in Discourse and Dialogue. pp.85-112.

Chanlekha H. and Kawtrakul A. 2004. Thai Named Entity Extraction by incorporating Maximum Entropy Model with Simple Heuristic Information. IJCNLP' 2004.

Chareonsuk J ., Sukvakree T., and Kawtrakul A. 2005. Elementary Discourse unit Segmentation for

Thai using Discourse Cue and Syntactic Information. NCSEC 2005.

Chang D.S. and Choi K.S. 2004. Causal Relation Extraction Using Cue Phrase and Lexical Pair Probabilities. IJCNLP. pp. 61 – 70.

Charniak, E. 2000. A maximum-entropy-inspired parser. Proc. of NAACL, pp.132-130.

Girju R. and Moldovan D. 2002. Mining answers for causation questions. AAAI Symposium on Mining Answers from Texts and Knowledge Bases.

Inui T., Inui K., and Matsumoto Y. 2004. Acquiring causal knowledge from text using the connective markers. Journal of the information processing society of Japan 45(3), pp. 919-993.

Lemeire, J., S. Maes and E. Dirkx. 2004. Causal Models for Parallel Performance Analysis. Fourth PA3CT-Symposium, Edegem, Belgium, September.

Marcu D. and Echihabi A. 2002. An Unsupervised Approach to Recognizing Discourse Relations. in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics Conference. pp. 368 – 375.

Pechsiri C., Kawtrakul A. and Piriyakul R. 2005. Mining Causality Knowledge From Text for Question Answering System. IEICE Transactions on Information and Systems, Vol.E90-D, No.10 :1523-1533.

Pengphon N., Kawtrakul A., and Suktarachan M. 2002. Word Formation Approach to Noun Phrase Analysis for Thai. SNLP.

Mitchell T.M. 1997. Machine Learning. The McGraw-Hill Companies Inc. and MIT Press, Singapore.

Sudprasert S. and Kawtrakul A. 2003. Thai Word Segmentation based on Global and Local Unsupervised Learning. NCSEC'2003.

Talmy, L. 2000. Toward a Cognitive Semantics Concept Structuring Systems – Vol. 1. The MIT Press.

Torisawa K. 2003. Automatic Extraction of Commonsense Inference Rules from Corpora. In Proc. Of The 9th Annual Meeting of The Association for Natural Language Proceeding. pp. 318-321.

Trnkova, Jana, Wolfgang Theilmann. 2004. Authoring processes for Advanced Learning Strategies. Telecooperation Research Group,TU Darmstadt, and SAP Research, CEC Karlsruhe. Germany.

Wolff, P. 2007. Representing Causation. Journal of experimental psychology: General 2007 Vol. 136 No.1 82-111. USA.

Context Modeling for IQA: The Role of Tasks and Entities

Raffaella Bernardi and **Manuel Kirschner**
KRDB, Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
{bernardi, kirschner}@inf.unibz.it

Abstract

In a realistic Interactive Question Answering (IQA) setting, users frequently ask follow-up questions. By modeling how the questions' focus evolves in IQA dialogues, we want to describe what makes a particular follow-up question salient. We introduce a new focus model, and describe an implementation of an IQA system that we use for exploring our theory. To learn properties of salient focus transitions from data, we use logistic regression models that we validate on the basis of predicted answer correctness.

1 Questions within a Context

Question Answering (QA) systems have reached a high level of performance within the scenario originally described in the TREC competitions, and are ready to tackle new challenges as shown by the new tracks proposed in recent instantiations (Voorhees, 2004). To answer these challenges, attention is moving towards adding semantic information at different levels. Our work is about context modeling for Interactive Question Answering (IQA) systems. Our research hypothesis is that a) knowledge about the dialogue history, and b) lexical knowledge about semantic arguments improve an IQA system's ability to answer follow-up questions. In this paper we use logistic regression modeling to verify our claims and evaluate how the performance of our Q→A mapping algorithm varies based on whether such knowledge is taken into account.

Actual IQA dialogues often exhibit "context-dependent" follow-up questions (FU Qs) containing anaphoric devices, like Q2 below. Such questions are potentially difficult to process by means of standard QA techniques, and it is for these cases that we claim that predicting the FU question's focus (here, the entity "library card") will help a system find the correct answer (cf. Sec. 6 for empirical backup).

Q1: Can high-school students use the library?
A1: Yes, if they got a library card.
Q2: So, how do I get it?

Following (Stede and Schlangen, 2004), we refer to the type of IQA dialogues we are studying as "information-seeking chat", and conjecture that this kind of dialogue can be handled by means of a simple model of discourse structure. Our assumption is that in general the user engages in a coherent dialogue with the system. As proposed in (Ahrenberg et al., 1995), we model the dialogues in terms of pairs of initiatives (questions) and responses (answers), ignoring other intentional acts.

The approach we adopt aims at answering the following questions: (a) In what way does information about the previous user questions and previous system answers help in predicting the next FU Q? (b) Does the performance of an IQA system improve if it has structure/history-based information? (c) Which is the role that each part of this information plays for determining the correct answer to a FU Q?

This paper is structured as follows. Section 2 gives an overview of some theories of focus used in dialogue and IQA. Section 3 then gives a detailed account of our theory, explaining *what* a question can focus on, and what patterns of focus change we expect a FU Q will trigger. Hence, this first

Coling 2008: Proceedings of the workshop on Knowledge and Reasoning for Answering Questions, pages 25–32
Manchester, August 2008

part answers our question (a) above. We then move to more applied issues in Sec. 4, where we show how questions and answers were annotated with focus information. The next Section 5 explains the Q→A algorithm we use to test our theory so as to answer (b), while Section 6 covers the logistic regression models with which we learn optimal values for the algorithm from data, addressing question (c).

2 Coherence in IQA dialogues

In the area of Discourse processing, much work has been devoted to formulating rules that account for the coherence of dialogues. This coherence can often be defined in terms of *focus* and *focus shifts*. In the following, we adopt the definition from (Lecœuche et al., 1999): *focus* stands for the "set of all the things to which participants in a dialogue are attending to at a certain point in a dialogue".[1] In general, all theories of dialogue focus considered by Lecœuche *et al.* claim that the focus changes according to some specific and well defined patterns, following the rules proposed by the respective theory. The main difference between these theories lies in how these rules are formulated.

A major distinguishing feature of different focus theories has been the question whether they address global or local focus. While the latter explain coherence between consecutive sentences, the former are concerned with how larger parts of the dialogue can be coherent. We claim that in "information seeking dialogue" this distinction is moot, and the two kinds of foci collapse into one. Furthermore, our empirical investigation shows that it suffices to consider a rather short history of the dialogue, i.e. the previous user question and previous system answer, when looking for relations between previous dialogue and a FU Q.

Salient transitions between two consecutive questions are defined in (Chai and Jin, 2004) under the name of "informational transitions". The authors aim to describe how the topic within a dialogue evolves. They take "entities" and "activities" as the main possible focus of a dialogue. A FU Q can be used to ask (i) a similar question as the previous one but with different constraints or different participants (topic extension); (ii) a question concerning a different aspect of the same topic (topic exploration); (iii) a question concerning a related activity or a related entity (topic shift). We take this analysis as our starting point, extend it and propose an algorithm to automatically detect the kind of focus transition a user performs when asking a FU Q, and evaluate our extended theory with real dialogue data. Following (Bertomeu et al., 2006) we consider also the role of the system answer, and we analyze the thematic relations between the current question and previous question, and the current question and previous answer. Unlike (Bertomeu et al., 2006), we attempt to learn a model of naturally occurring thematic relations in relatively unconstrained IQA dialogues.

3 Preliminary Observations

3.1 What "things" do users focus on?

For all forthcoming examples of dialogues, questions and answers, we will base our discussion on an actual prototype IQA system we have been developing; this system is supposed to provide library-related information in a university library setting.

In the dialogues collected via an earlier Wizard-of-Oz (WoZ) experiment (Kirschner and Bernardi, 2007), we observed that users either seem to have some specific library-related *task* (action, e.g. "search") in mind that they want to ask the system about, or they want to retrieve information on some specific *entity* (e.g., "guided tour"). People tend to use FU Qs to "zoom into" (i.e., find out more about) either of the two. In line with this analysis, the focus of a FU Q might move from the task (action/verb) to the entities that are possible fillers of the verb's semantic argument slots.

Based on these simple observations, we propose a task/entity-based model for describing the focus of questions and answers in our IQA setting. Our theory of focus structure is related to the task-based theory of (Grosz, 1977). Tasks correspond to verbs, which are inherently connected to an argument structure defining the verb's semantic roles. By consulting lexical resources like PropBank (Palmer et al., 2005), we can use existing knowledge about possible semantic arguments of

[1] This definition is in line with how *focus* has been used in Computational Linguistics and Artificial Intelligence (hence, "AI focus"), originating in the work of Grosz and Sidner on discourse entity salience. We follow Lecœuche *et al.* in that focused elements could also be actions/tasks. We see the most salient focused element (corresponding to the "Backward-looking center" in Centering Theory) as the *topic* of the utterance. Accordingly, in the following we will use the terms *focus* and *topic* interchangeably; cf. (Vallduvi, 1990) for a survey of these rather overloaded terms.

the tasks we have identified.

We claim that actions/verbs form a suitable and robust basis for describing the (informational) meaning of utterances in IQA. Taking the main verb along with its semantic arguments to represent the core meaning of user questions seems to be a more feasible alternative to deep semantic approaches that still lack the robustness for dealing with unconstrained user input.

Further, we claim that analyzing user questions on the basis of their task/entity structure provides a useful level of abstraction and granularity for empirically studying informational transitions in IQA dialogues. We back up this claim in Section 6. Along the lines of (Kirschner and Bernardi, 2007), we aim for a precise definition of focus structure for IQA questions. Our approach is similar in spirit to (Chai and Jin, 2004), whereas we need to reduce the complexity of their discourse representation (i.e., their number of possible question "topics") so that we arrive at a representation of focus structure that lends itself to implementation in a practical IQA system.

3.2 How focus evolves in IQA

We try to formulate our original question, "Given a user question and a system response, what does a salient FU Q focus on?" more precisely. We want to know whether the FU Q initiates one of the following three transitions:[2]

Topic zoom asking about a different aspect of what was previously focused

1. asking about the same task and same argument, but different question type (e.g., search for books: Q: where, FU Q: how)
2. asking about the same entity (e.g., guided tour: Q: when, FU Q: where)
3. asking about the same task but different argument (e.g., Q: search for books, FU Q: search for journals)
4. asking about an entity introduced in the *previous system answer*

Coherent shift to a "related" (semantically, or: verb→its semantic argument) focus

1. from task to semantically related task
2. from task to related entity: entity is a semantic argument of the task
3. from entity to semantically related entity
4. from entity to related task: entity is a semantic argument of the task

Shift to an unrelated focus

From the analysis of our WoZ data we get certain intuitions about salient focus flow between some preceding dialogue and a FU Q. First of all, we learn that a dialogue context of just one previous user question and one previous system answer generally provides enough information to resolve context-dependent FU Qs. In the remainder of this section, we describe the other intuitions by proposing alternative ways of detecting the focus of a FU Q that follows a salient relation ("Topic zoom" or "Coherent shift"). Later in this paper we show how we implement these intuitions as features, and how we use a regression model to learn the importance of these features from data.

Exploiting task/entity structure Knowing which entities are possible semantic arguments of a library-related task can help in detecting the focused task. Even if the task is not expressed explicitly in the question, the fact that a number of participant entities *are* found in the question could help identify the task at hand.

Exploiting (immediate) dialogue context: previous user question It might prove useful to know the things that the immediately preceding user question focused on. If users tend to continue focusing on the same task, entity or question type, this focus information can help in "completing" context-dependent FU Qs where the focused things cannot be detected easily since they are not mentioned explicitly. This way of using dialogue context has been used in previous IQA systems, e.g., the Ritel system (van Schooten et al., forthcoming).

Exploiting (immediate) dialogue context: previous system answer Whereas the role of the system answer has been ignored in some previous accounts of FU Qs (e.g., (Chai and Jin, 2004) and even in the highly influential TREC task (Voorhees, 2004)), our data suggest that the system answer does play a role for predicting what a FU Q will focus on: it seems that the system answer can introduce entities that a salient FU Q will ask more information about. (van Schooten and op den Akker, 2005) and (Bertomeu et al., 2006) describe IQA systems that also consider the previous system answer.

[2] Comparing our points to (Chai and Jin, 2004), Topic zoom: 1. and 2. are cases of topic exploration, 3. of topic extension, and 4. is new. Coherent shift: 1. and 2. are cases of topic shift, and 3. and 4. are new.

Exploiting task/entity structure combined with dialogue context It might be useful to combine knowledge about the task/entity structure with knowledge about the previously focused task or entity. E.g., a previously focused task might make a "coherent shift" to a participant entity likely; likewise, a previously focused entity might enable a coherent shift to a task in which that entity could play a semantic role.

The questions to be addressed in the remainder of the paper now are the following. Does the performance of an IQA system improve if it has structure/history-based information as mentioned above? Which is the role that each part of this information plays for determining the correct answer to a FU Q?

4 Tagging focus on three levels

Following the discussion in Section 3.1, and having studied the user dialogues from our WoZ data, we propose to represent the (informational) meaning of a user question by identifying the task and/or entity that the question is about (*focuses on*). Besides task and entity, we have Question Type (QType) as a third level on which to describe a question's focus. The question type relates to what type of information the user asks about the focused task/entity, and equivalently describes the exact *type of answer* (e.g., why, when, how) that the user hopes to get about the focused task/entity. Thus, we can identify the focus of a question with the triple <Task, Entity, QType>.

We have been manually building a small domain-dependent lexical resource that in the following we will call "task/entity structure". We see it as a miniature version of the PropBank, restricted to the small number of verbs/tasks that we have identified to be relevant in our domain, but extended with some additional semantic argument slots if required. Most importantly, the argument slots have been assigned to possible filler entities, each of which can be described with a number of synonymous names.

Tasks By analyzing a previously acquired extensive list of answers to frequently-asked library-related questions, we identified a list of 11 tasks that library users might ask about (e.g. search, reserve, pick up, browse, read, borrow, etc.). Our underlying assumption is that the focus (as identified by the focus triple) of a question is identical to that of the corresponding answer. Thus, we assume the focus triple describing a user question also describes its correct answer. For example, in Table 1, A1 would share the same focus triple as Q1.

We think of the tasks as abstract descriptions of actions that users can perform in the library context. A user question *focuses on* a specific task if it either explicitly contains that verb (or a synonym), or implicitly refers to the same "action frame" that the verb instantiates.

Entities Starting from the information about semantic arguments of these verbs available in PropBank, and extending it when necessary for domain-specific use of the verbs, for each task we determined its argument slots. Again by inspecting our list of FAQ answers, we started assigning library-related entities to these argument slots, when we found that the answer focuses on both the task and the *semantic argument* entity. We found that many answers focus on some library-related entity without referring to any task. Thus, we explicitly provide for the possibility of a question/answer being about just an entity, e.g.: "What are the opening times?". A user question focuses on a specific entity if it refers to it explicitly or via some reference phenomenon (anaphora, ellipsis, etc.) linked to the dialogue history.

Question Types We compiled a list of question (or answer) types by inspecting our FAQ answers list, and thinking about the types of questions that could have given rise to these answers. We aimed for a compromise between potentially more fine-grained distinctions of question semantics, and better distinguishability of the resulting set of labels (for a human annotator or a computer program).

We defined each question type by providing a typical question template, e.g.: "where: where can I find $Entity?", "whatis: what is $Entity?", "yesno: can I $Task $Entity?', "howto: how do I $Task $Entity?". Note how some question types capture questions that focus on some task along with some participant entity, while others focus on just an entity. We also devised some question types for questions focusing on just a task, where we assume an *implicit* semantic argument which is not expressed, e.g., "how can I borrow?" (where in the specific context of our application we can imply a semantic argument like "item"). A question has a specific question type if it can be paraphrased with the corresponding question template. An answer

has a specific type if it is the correct answer to that question template.

4.1 A repository of annotated answers

From our original collection of answers to library FAQs, we have annotated around 200 with focus triples. The triples we selected include all potential answers to the FU Qs from the free FU Q elicitation experiment described in the next section. Some of the actual answers were annotated with more than one focus triple, e.g., often the answer corresponded to more than one question type. The total of 207 focus triples include all 11 tasks and 23 different question types (where the 4 most frequent types were the ones mentioned as examples above, accounting for just over 50% of all focus triples).

For instance, the answer: "You can restrict your query in the OPAC on individual Library locations. The search will then be restricted e.g. to the Library of Bressanone-Brixen or the library of the 'Museion'." is marked by: <Task: search, Entity: specific library location, QType: yesno>.

The algorithm we introduce in Section 5 uses this answer repository as the set A of potential candidates from which it chooses the answer to a new user question. Again, we assume that if we can determine the correct focus triple of a user question, the answer from our collection that has been annotated with that same triple will correctly answer the question.

4.2 Annotated user questions

Having created an answer repository annotated with focus triples, we need user questions annotated on the same three levels, which we can then use for training and evaluating the Q→A algorithm that we introduce in Section 5. We acquired these data in two steps: 1. eliciting free FU Qs from subjects in a web-based experiment, 2. annotating the questions with focus triples.

Dialogue Collection Experiment We set up a web-based experiment to collect genuine FU Qs. We adopted the experimental setup proposed in (van Schooten and op den Akker, 2005)), in that we presented to our subjects short dialogues consisting of a first library-related question, and a corresponding correct answer, as exemplified by "Q1" and "A1" in Table 1.

We asked the subjects to provide a FU Q "Q2" such that it will help further serve their information need in the situation defined by the given previous question-answer exchange. In this way, we collected 88 FU Qs from 8 subjects and 11 contexts (first questions and answers).[3]

Annotating the questions We annotated these 88 FU Qs, along with the 11 first questions that were presented to the subjects, with focus triples. By (informally) analyzing the differences between different annotators' results, we continuously tried to disambiguate and improve the annotation instructions. As a result, we present a pre-compiled list of entities from which the annotator selects the one they consider to be in focus, and that of all possible candidates is the one least "implied" by the context. Table 1 shows one example annotation of one of the 11 first user questions and two of the 8 corresponding FU Qs.

5 A feature-based Q→A algorithm

We now present an algorithm for mapping a user question to a canned-text answer from our answer repository. The decision about which answer to select is based on a score that the algorithm assigns to each answer, which in turn depends on the values of the features we have introduced in the previous section. Thus, the purpose of the algorithm is to select the best answer focus triple from the repository, based on feature values. In this way, we can use the algorithm as a test bed for identifying features that are good indicators for a correct answer. Our goal is to evaluate the algorithm based on its accuracy in finding correct focus triples (which are the "keys" to the actual system answers) for user questions (see Section 5.2).

For each new user question q that is entered, the algorithm iterates through all focus triples a in the annotated answer repository A (cf. Section 4.1). For each combination of q and a, all 10 features $x_{1,q,a} \ldots x_{10,q,a}$ are evaluated. Each feature that evaluates to true ($\beta = 1$) or some positive value, contributes with this score β towards the overall score of a. The algorithm then returns the highest-scoring answer $\hat{a}$.

$$\hat{a} = \arg\max_{a \in A}(\beta_1 x_{1,q,a} + \cdots + \beta_{10} x_{10,q,a})$$

[3] In the future, we plan to collect real FU Qs from users of our online IQA system, which will solve the potential problem of these questions being somewhat artificial due to the experimental setting. However, we still expect our current data to be highly relevant for studying what users would ask about next.

ID	Q/A	Task	Entity	QType
Q1	Can I get search results for a specific library location?	search	specific library location	yesno
A1	You can restrict your query in the OPAC on individual Library locations. (...)			
Q2a	How can I do that?	search	specific library location	howto
Q2b	How long is my book reserved there if I want to get it?	reserve	my book	howlong

Table 1: Example annotation of one first question and two corresponding FU Qs

5.1 Features

Based on the intuitions presented in Section 3.2, we now describe the 10 features $x_{1,q,a}, \ldots, x_{10,q,a}$ that our algorithm uses as predictors for answer correctness. All Task and Entity matching is done using string matching over word stems. QType matching uses regular expression matching with a set of simple regex patterns we devised for our QTypes.

- 3 surface-based features $x_{1,q,a}, \ldots, x_{3,q,a}$: whether $\{\text{Task}_a, \text{Entity}_a, \text{QType}_a\}$ are matched in q. Entity feature returns the length in tokens of the matched entity.
- 1 task/entity structure-based feature $x_{4,q,a}$: how many of the participant entities of Task_a (as encoded in our task/entity structure) are matched in q.
- 4 focus continuity features $x_{5,q,a}, \ldots, x_{8,q,a}$: whether $\{\text{Task}_a, \text{Entity}_a, \text{QType}_a\}$ are continued in q, wrt. previous dialogue as follows:[4]
  - Task, Entity, QType continuity wrt. previous user question.
  - Entity continuity wrt. previous system answer.
- 2 task/entity structure + focus continuity features $x_{9,q,a}, x_{10,q,a}$:
  - Focused Task of previous user question has Entity_a as a participant.
  - Task_a has focused Entity of previous question as a participant.

5.2 First Evaluation

Table 2 shows manually set feature scores $\beta_1, \ldots, \beta_{10}$ we used for a first evaluation of the algorithm; we chose these particular scores after inspecting our WoZ data. With these scores, we ran the Q→A algorithm on the annotated questions of annotator 1, who had provided a "gold standard" annotation for 78 of the 99 user questions (the remainder of the questions are omitted because the annotator did not know how to assign a focus triple to them). For 24 out of 78 questions, the algorithm found the exact focus triple (from a total of 207 focus triples in the answer repository), yielding an accuracy of 30.8%.

k	$x_{k,q,a}$	$\text{range}(x_{k,q,a})$	β_k
1	qTypeMatch	0,1	4
2	taskMatch	0,1	3
3	lenEntityMatch	n	2
4	nEntitiesInTask	n	1
5	taskContinuity	0,1	1
6	entityContinuity	0,1,2	1
7	qTypeContinuity	0,1	1
8	entityInPrevAnsw	0,1,2	2
9	entityInPrevTask	0,1	1
10	prevEntityInTask	0,1	1

Table 2: Manually set feature scores

6 Logistic Regression Model

To improve the accuracy of the Q→A algorithm and to learn about the importance of the single features for predicting whether an answer from A is correct, we want to learn optimal scores $\beta_1, \ldots, \beta_{10}$ from data. We use a logistic regression model (cf. (Agresti, 2002)). Logistic regression models describe the relationship between some predictors (i.e., our features) and an outcome (answer correctness).

We use the logit β coefficients $\beta_1, \ldots, \beta_k$ that the logistic regression model estimates (from training data, using maximum likelihood estimation)

[4] Both entity continuity features evaluate to '2' when exactly the same entity is used again, but to '1' when a synonym of the first entity is used.

	Coeff.	95% C.I.
lenEntityMatch	6.76	5.26–8.26
qTypeMatch	2.54	2.02–3.06
taskContinuity	2.17	1.39–2.94
entityInPrevAnsw	1.78	1.06–2.49
taskMatch	1.37	0.80–1.94
prevEntityInTask	-1.24	-2.06– -0.43

Table 3: Model M_2: Magnitudes of significant effects

for the predictors as empirically motivated scores. In contrast to other supervised machine learning techniques, regression models yield human-readable coefficients that show the individual effect of each predictor on the outcome variable.

6.1 Generating Training data

We generate the training data for learning the logistic regression model from our annotated answer repository A (Sec. 4.1) and annotated questions (Sec. 4.2) as follows. For each human-annotated question q and each candidate answer focus triple from our repository ($a \in A$), we evaluate our features $x_{1,q,a}, \ldots, x_{10,q,a}$. If the focus triples of q and a are identical, we take the particular feature values as a training instance for a correct answer; if the focus triples differ, we have a training instance for a wrong answer.[5]

6.2 Results and interpretation

We fit model M_1 based on the annotation of annotator 2 using all 10 features.[6] We then fit a second model M_2, this time including only the 6 features that correspond to coefficients from model M_1 that are significantly different from zero. Table 3 shows the resulting logit β coefficients with their 95% confidence intervals. Using these coefficients as new scores in our Q→A algorithm (and setting all non-significant coefficients' feature scores to 0), it finds the correct focus triple for 47 out of 78 test questions (as before, annotated by annotator 1); answer accuracy now reaches 60.3%.

We interpret the results in Table 3 as follows. All three surface-based features are significant predictors of a correct answer. The length of the matched entity contributes more than the other two; we attribute this to the fact that there are more cases where our simple implementations of qTypeMatch and taskMatch fail to detect the correct QType or task. While the task/entity structure-based nEntitiesInTask clearly misses to reach significance, the history-based features taskContinuity and entityInPrevAnsw are useful indicators for a correct answer. The first is evidence for "Topic zoom", with the FU Q asking about a different aspect of the previously focused task, while the second shows the influence of the previous answer in shaping the entity focus of the FU Q. From the two "task/entity structure + focus continuity" features, we find that if a FU Q focuses on a task that in our task/entity structure has an argument slot filled with the previously focused entity, it actually indicates a *false* answer; the implications of this finding will have to be explored in future work.

Finally, to pinpoint the important contributions of structure- and/or focus continuity features, we fit a new model M_3, this time including only the 3 (significant) surface-based features. Evaluating the resulting coefficients in the same way as above, we get only 24 out of 78 correct answer focus triples, an accuracy of 30.8%. This result supports our initial claim that an IQA system improves if it has a way of predicting the focus of a FU Q.

7 Conclusion

Our original hypothesis was that a) knowledge about the dialogue history, and b) lexical knowledge about semantic arguments could improve an IQA system's ability to answer FU Qs. We operationalized these notions by formulating a set of 10 features that evaluate whether a candidate answer is the correct one given a new (FU) user question. We then used regression modeling to investigate the usefulness of each individual feature by learning from annotated IQA dialogue data, showing that certain knowledge about the dialogue history (the previously focused task, and the entities mentioned in the previous system answer) and about semantic arguments are useful for distinguishing correct from wrong answers to a FU Q. Finally, we evaluated these results by showing how our Q→A mapping algorithm's answer accuracy improved by using the empirically learned scores for all statistically significant predictors/features. The features and the Q→A algorithm as a whole are based on a simple way to describe IQA questions

[5] Although in this way we get imbalanced data sets with $|A| - 1$ negative training instances for each positive one, we have not yet explored this issue further.

[6] We use annotator 2's data for training, and annotator 1's for testing throughout this paper.

in terms of focus triples. By showing how we have improved an actual system with learned feature scores, we demonstrated this representation's viability for implementation and for empirically studying informational transitions in IQA.

Although the IQA system used in our project is in several ways limited, our findings about how focus evolves in real IQA dialogues should scale up to any new or existing IQA system that allows users to ask context-dependent FU Qs in a type of "information seeking" paradigm. It would be interesting to see how this type of knowledge could be added to other IQA or dialogue systems in general.

We see several directions for future work. Regarding coherent focus transitions, we have to look into which transitions to different tasks/entities are more coherent than others, possibly based on semantic similarity. A major desideratum for showing the scaleability of our work is to explore the influence of the subjects on our data annotation. We are currently working on getting an objective inter-annotator agreement measure, using external annotators. Finally, we plan to collect a large corpus of IQA dialogues via a publicly accessible IQA system, and have these dialogues annotated. With more data, coming from genuinely interested users instead of experimental subjects, and having these data annotated by external annotators, we expect to have more power to find significant and generally valid patterns of how focus evolves in IQA dialogues.

Acknowledgments

We thank Marco Baroni, Oliver Lemon, Massimo Poesio and Bonnie Webber for helpful discussions.

References

Agresti, Alan. 2002. *Categorical Data Analysis*. Wiley-Interscience, New York.

Ahrenberg, L., N. Dahlbäck, and A. Jönsson. 1995. Coding schemes for studies of natural language dialogue. In *Working Notes from AAAI Spring Symposium*, Stanford.

Bertomeu, Núria, Hans Uszkoreit, Anette Frank, Hans-Ulrich Krieger, and Brigitte Jörg. 2006. Contextual phenomena and thematic relations in database QA dialogues: results from a wizard-of-oz experiment. In *Proc. of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 1–8, New York, NY.

Chai, Joyce Y. and Rong Jin. 2004. Discourse structure for context question answering. In *Proc. of the HLT-NAACL 2004 Workshop on Pragmatics in Question Answering*, Boston, MA.

Grosz, Barbara Jean. 1977. *The representation and use of focus in dialogue understanding.* Ph.D. thesis, University of California, Berkeley.

Kirschner, Manuel and Raffaella Bernardi. 2007. An empirical view on iqa follow-up questions. In *Proc. of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium.

Lecœuche, Renaud, Chris Mellish, Catherine Barry, and Dave Robertson. 1999. User-system dialogues and the notion of focus. *Knowl. Eng. Rev.*, 13(4):381–408.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106.

Stede, Manfred and David Schlangen. 2004. Information-seeking chat: Dialogue management by topic structure. In *Proc. of SemDial'04 (Catalog)*, Barcelona, Spain.

Vallduvi, Enric. 1990. *The Informational Component*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

van Schooten, Boris and Rieks op den Akker. 2005. Follow-up utterances in QA dialogue. *Traitement Automatique des Langues*, 46(3):181–206.

van Schooten, Boris, R. op den Akker, R. Rosset, O. Galibert, A. Max, and G. Illouz. forthcoming. Follow-up question handling in the IMIX and Ritel systems: A comparative study. *Journal of Natural Language Engineering*.

Voorhees, Ellen M. 2004. Overview of the TREC 2004 question answering track. In *Proc. of the 13th Text REtrieval Conference*.

Personalized, Interactive Question Answering on the Web

Silvia Quarteroni
University of Trento
Via Sommarive 14
38100 Povo (TN), Italy
silviaq@disi.unitn.it

Abstract

Two of the current issues of Question Answering (QA) systems are the lack of personalization to the individual users' needs, and the lack of interactivity by which at the end of each Q/A session the context of interaction is lost.

We address these issues by designing and implementing a model of personalized, interactive QA based on a User Modelling component and on a conversational interface. Our evaluation with respect to a baseline QA system yields encouraging results in both personalization and interactivity.

1 Introduction

Information overload, i.e. the presence of an excessive amount of data from which to search for relevant information, is a common problem to Information Retrieval (IR) and its subdiscipline of Question Answering (QA), that aims at finding concise answers to questions in natural language. In Web-based QA in particular, this problem affects the relevance of results with respect to the users' needs, as queries can be ambiguous and even answers extracted from documents with relevant content but expressed in a difficult language may be ill-received by users.

While the need for user personalization has been addressed by the IR community for a long time (Belkin and Croft, 1992), very little effort has been carried out up to now in the QA community in this direction. Indeed, personalized Question Answering has been advocated in TREC-QA starting from 2003 (Voorhees, 2003); however, the issue was solved rather expeditiously by designing a scenario where an "average news reader" was imagined to submit the 2003 task's *definition* questions.

Moreover, a commonly observed behavior in users of IR systems is that they often issue queries not as standalone questions but in the context of a wider information need, for instance when researching a specific topic. Recently, a new research direction has been proposed, which involves the integration of QA systems with dialogue interfaces in order to encourage and accommodate the submission of multiple related questions and handle the user's requests for clarification in a less artificial setting (Maybury, 2002); however, Interactive QA (IQA) systems are still at an early stage or applied to closed domains (Small et al., 2003; Kato et al., 2006). Also, the "complex, interactive QA" TREC track (`www.umiacs.umd.edu/~jimmylin/ciqa/`) has been organized, but here the interactive aspect refers to the evaluators being enabled to interact with the systems rather than to dialogue *per se*.

In this paper, we first present an adaptation of User Modelling (Kobsa, 2001) to the design of personalized QA, and secondly we design and implement an interactive open-domain QA system, YourQA. Section 2 briefly introduces the baseline architecture of YourQA. In Section 3, we show how a model of the user's reading abilities and personal interests can be used to efficiently improve the quality of the information returned by a QA system. We provide an extensive evaluation methodology to assess such efficiency by improving on our previous work in this area (Quarteroni and Manandhar, 2007b).

Moreover, we discuss our design of interactive QA in Section 4 and conduct a more rigorous evaluation of the interactive version of YourQA by comparing it to the baseline version on a set of TREC-QA questions, obtaining encouraging results. Finally, a unified model of personalized, interactive QA is described in Section 5.

2 Baseline System Architecture

The baseline version of our system, YourQA, is able to extract answers to both factoid and non-factoid questions from the Web. As most QA systems (Kwok et al., 2001), it is organized according to three phases:

- **Question Processing**: The query is classified and the two top expected answer types are estimated; it is then submitted to the underlying search engine;
- **Document Retrieval**: The top n documents are retrieved from the search engine (Google, `www.google.com`) and split into sentences;

Coling 2008: Proceedings of the workshop on Knowledge and Reasoning for Answering Questions, pages 33–40
Manchester, August 2008

- **Answer Extraction**:
    1. A sentence-level similarity metric combining lexical, syntactic and semantic criteria is applied to the query and to each retrieved document sentence to identify candidate answer sentences;
    2. Candidate answers are ordered by relevance to the query; the Google rank of the answer source document is used as a tie-breaking criterion.
    3. The list of top ranked answers is then returned to the user in an HTML page.

Note that our answers are in the form of sentences with relevant words or phrases highlighted (as visible in Figure 2) and surrounded by their original passage. This is for two reasons: we believe that providing a context to the exact answer is important and we have been mostly focusing on non-factoids, such as definitions, which it makes sense to provide in the form of a sentence. A thorough evaluation of YourQA is reported in e.g. (Moschitti et al., 2007); it shows an F1 of 48±.7 for non-factoids on Web data, further improved by a SVM-based re-ranker.

In the following sections, we describe how the baseline architecture is enhanced to accommodate personalization and interactivity.

3 User Modelling for Personalization

Our model of personalization is centered on a User Model which represents students searching for information on the Web according to three attributes:

1. age range $a \in \{7-10, 11-16, adult\}$,
2. reading level $r \in \{basic, medium, advanced\}$;
3. profile p, a set of textual documents, bookmarks and Web pages of interest.

Users' age[1] and browsing history are typical UM components in news recommender systems (Magnini and Strapparava, 2001); personalized search systems such as (Teevan et al., 2005) also construct UMs based on the user's documents and Web pages of interest.

3.1 Reading Level Estimation

We approach reading level estimation as a supervised learning task, where representative documents for each of the three UM reading levels are collected to be labelled training instances and used to classify previously unseen documents.

Our training instances consist of about 180 HTML documents from a collection of Web portals[2] where pages are explicitly annotated by the publishers according to the three reading levels above. As a learning model, we use unigram language modelling introduced in (Collins-Thompson and Callan, 2004) to model the reading level of subjects in primary and secondary school.

Given a set of documents, a unigram language model represents such a set as the vector of all the words appearing in the component documents associated with their corresponding probabilities of occurrence within the set.

In the test phase of the learning process, for each unclassified document D, a unigram language model is built (as done for the training documents). The estimated reading level of D is the language model lm_i maximizing the likelihood that D has been generated by lm_i (In our case, three language models lm_i are defined, where $i \in \{basic, medium, advanced\}$.). Such likelihood is estimated using the function:

$$L(lm_i|D) = \sum_{w \in D} C(w, D) \cdot log[P(w|lm_i)], \quad (1)$$

where w is a word in the document, $C(w, d)$ represents the number of occurrences of w in D and $P(w|lm_i)$ is the probability that w occurs in lm_i (approximated by its frequency).

3.2 Profile Estimation

Information extraction from the user's documents as a means of representation of the user's interests, such as his/her desktop files, is a well-established technique for personalized IR (Teevan et al., 2005).

Profile estimation in YourQA is based on key-phrase extraction, a technique previously employed in several natural language tasks (Frank et al., 1999).

For this purpose, we use Kea (Witten et al., 1999), which splits documents into phrases and chooses some of the phrases as be key-phrases based on two criteria: the first index of their occurrence in the source documents and their $TF \times IDF$ score[3] with respect to the current document collection. Kea outputs for each document in the set a ranked list where the candidate key-phrases are in decreasing order; after experimenting with several values, we chose to use the top 6 as key-phrases for each document.

The profile resulting from the extracted key-phrases is the base for all the subsequent QA activity: any question the user submits to the QA system is answered by taking such profile into account, as illustrated below.

3.3 Personalized QA Algorithm

The interaction between the UM component and the core QA component modifies the standard QA process at the Answer Extraction phase, which is modified as follows:

[1] Although the reading level can be modelled separately from the age range, for simplicity we here assume that these are paired in a reading level component.

[2] Such Web portals include: `bbc.co.uk/schools`, `www.think-energy.com`, `kids.msfc.nasa.gov`.

[3] The $TF \times IDF$ of a term t in document D within a collection S is: $TF \times IDF(t, D, S) = P(t \in D) \times -logP(t \in [S/D])$.

1. The retrieved documents' reading levels are estimated;
2. Documents having a different reading level from the user are discarded; if the remaining documents are insufficient, part of the incompatible documents having a close reading level are kept;
3. From the documents remaining from step 2, key-phrases are extracted using Kea;
4. The remaining documents are split into sentences;
5. Document topics are matched with the topics in the UM that represent the user's interests;
6. Candidate answers are extracted from the documents and ordered by relevance to the query;
7. As an additional answer relevance criterion, the degree of match between the candidate answer document topics and the user's topics of interest is used and a new ranking is computed on the initial list of candidate answers.

Step 7 deserves some deeper explanation. For each document composing the UM profile and the retrieved document set, a ranked list of key-phrases is available from the previous steps. Both key-phrase sets are represented by YourQA as arrays, where each row corresponds to one document and each column corresponds to the rank within such document of the key-phrase in the corresponding cell.

As an illustrative example, a basic user profile, created from two documents about Italian cuisine and the movie "Ginger and Fred", respectively, might result in the following array:

$$\begin{bmatrix} pizza & lasagne & tiramisu & recipe & chef & egg \\ fred & ginger & film & music & movie & review \end{bmatrix}$$

The arrays of UM profile and retrieved document key-phrases are named P and $Retr$, respectively. We call $Retr_i$ the document represented in the i-th row in $Retr$, and P_n the one represented in the n-th row of P [4]. Given k_{ij}, i.e. the j-th key-phrase extracted from $Retr_i$, and P_n, i.e. the n-th document in P, we call $w(k_{ij}, P_n)$ the *relevance* of k_{ij} with respect to P_n. We define

$$w(k_{ij}, P_n) = \begin{cases} \frac{|Retr_i| - j}{|Retr_i|}, & k_{ij} \in P_n \\ 0, & otherwise \end{cases} \quad (2)$$

where $|Retr_i|$ is the number of key-phrases of $Retr_i$. The total relevance of document $Retr_i$ with respect to P, $w_P(Retr_i)$, is defined as the maximal sum of the relevance of its key-phrases, obtained for all the rows in P:

$$w_P(Retr_i) = max_{n \in P} \sum_{k_{ij} \in Retr_i} w(k_{ij}, P_n). \quad (3)$$

[4]Note that, while column index reflects a ranking based on the relevance of a key-phrase to its source document, row index only depends on the name of such document.

The personalized answer ranking takes w_P into account as a *secondary* ranking criterion with respect to the baseline system's similarity score; as before, Google rank of the source document is used as further a tie-breaking criterion.

Notice that our approach to User Modelling can be seen as a form of implicit (or quasi-implicit) relevance feedback, i.e. feedback not explicitly obtained from the user but inferred from latent information in the user's documents. Indeed, we take inspiration from (Teevan et al., 2005)'s approach to personalized search, computing the relevance of unseen documents (such as those retrieved for a query) as a function of the presence and frequency of the same terms in a second set of documents on whose relevance the user has provided feedback.

Our approaches to personalization are evaluated in Section 3.4.

3.4 Evaluating Personalization

The evaluation of our personalized QA algorithms assessed the contributions of the reading level attribute and of the profile attribute of the User Model.

3.4.1 Reading Level Evaluation

Reading level estimation was evaluated by first assessing the robustness of the unigram language models by running 10-fold cross-validation on the set of documents used to create such models, and averaging the ratio of correctly classified documents with respect to the total number of documents for each fold. Our results gave a very high accuracy, i.e. 94.23% $\pm$ 1.98 standard deviation.

However, this does not prove a direct effect on the user's perception of such levels. For this purpose, we defined *Reading level agreement (A_r)* as the percentage of documents rated by the users as suitable to the reading level to which they were assigned. We performed a second experiment with 20 subjects aged between 16 and 52 and with a self-assessed good or medium English reading level. They evaluated the answers returned by the system to 24 questions into 3 groups (basic, medium and advanced reading levels), by assessing whether they agreed that the given answer was assigned to the correct reading level.

Our results show that altogether, evaluators found answers appropriate for the reading levels to which they were assigned. The agreement decreased from 94% for A_{adv} to 85% for A_{med} to 72% for A_{bas}; this was predictable as it is more constraining to conform to a lower reading level than to a higher one.

3.4.2 Profile Evaluation

The impact of the UM profile was tested by using as a baseline the standard version of YourQA, where the UM component is inactive. Ten adult participants from various backgrounds took part in the experiment; they were invited to form an individual profile by brainstorming key-phrases for 2-3 topics of

their interest chosen from the Yahoo! directory (`dir.yahoo.com`): examples were "ballet", "RPGs" and "dog health".

For each user, we created the following 3 questions so that he/she would submit them to the QA system: Q_{per}, related to the user's profile, for answering which the *personalized* version of YourQA would be used; Q_{bas}, related to the user's profile, for which the *baseline* version of the system would be used; and Q_{unr}, unrelated to the user's profile, hence not affected by personalization. The reason why we handcrafted questions rather than letting users spontaneously interact with YourQA's two versions is that we wanted the results of the two versions to be different in order to measure a preference. After examining the top 5 results to each question, users had to answer the following questionnaire[5]:

- For each of the five results *separately*:

  TEST1: *This result is useful to me*:
  5) Yes, 4) Mostly yes, 3) Maybe, 2) Mostly not, 1) Not at all

  TEST2: *This result is related to my profile*:
  5) Yes, 4) Mostly yes, 3) Maybe, 2) Mostly not, 1) Not at all

- For the five results taken as a whole:

  TEST3: *Finding the info I wanted in the result page took*:
  1) Too long, 2) Quite long, 3) Not too long, 4) Quite little, 5) Very little

  TEST4: *For this query, the system results were sensitive to my profile*:
  5) Yes, 4) Mostly yes, 3) Maybe, 2) Mostly not, 1) Not at all

The experiment results are summarized in Table 1. The

Table 1: Profile evaluation results (avg $\pm$ st. dev.)

Measurement	Q_{rel}	Q_{bas}	Q_{unr}
TEST1	3.6±0.4	2.3±0.3	3.3±0.3
TEST2	4.0±0.5	2.2±0.3	1.7±0.1
TEST3	3.1±1.1	2.7±1.3	3.4±1.4
TEST4	3.9±0.7	2.5±1.1	1.8±1.2

first row reports a remarkable difference between the perceived usefulness for question Q_{rel} with respect to question Q_{bas} (answers to TEST1).

The results were compared by carrying out a one-way analysis of variance (ANOVA) and performing the Fischer test using the usefulness as factor (with the three queries as levels) at a 95% level of confidence. The test revealed an overall significant difference between factors, confirming that users are positively biased towards questions related to their own profile when it comes to perceived utility.

To analyze the answers to TEST2 (Table 1, row 2), which measured the perceived relatedness of each answer to the current profile, we used ANOVA again and and obtained an overall significant difference. Hence, answers obtained without using the users' profile were perceived as significantly less related to those obtained using their own profile, i.e. there is a significant difference between Q_{rel} and Q_{bas}. As expected, the difference between Q_{rel} and Q_{unr} is even more significant.

Thirdly, the ANOVA table computed using average perceived time (TEST3) as variable and the three questions as factors did not give any significance, nor did any of the paired t-tests computed over each result pair. We concluded that apparently, the time spent browsing results is not directly correlated to the personalization of results.

Finally, the average sensitivity of the five answers altogether (TEST4) computed over the ten participants for each query shows an overall significant difference in perceived sensitivity between the answers to question Q_{rel} (3.9±0.7) and those to question Q_{bas} (2.5±1.1) and Q_{unr} (1.8±1.2).

To conclude, our experience with profile evaluation shows that personalized QA techniques yield answers that are indeed perceived as more satisfying to users in terms of usefulness and relatedness to their own profile.

4 Interactivity

Making a QA system interactive implies maintaining and efficiently using the current dialogue context and the ability to converse with the user in a natural manner. Our implementation of IQA is guided by the following conversation scenario:

1. An optional reciprocal greeting, followed by a question *q* from the user;
2. *q* is analyzed to detect whether it is related to previous questions or not;
3. (a) If *q* is unrelated to the preceding questions, it is submitted to the QA component;
   (b) If *q* is related to the preceding questions (follow-up question), it is interpreted by the system in the context of previous queries; a revised version of *q*, *q'*, is either directly submitted to the QA component or a request for confirmation (grounding) is issued to the user; if he/she does not agree, the system asks the user to reformulate the question until it can be interpreted by the QA component;
4. As soon as the QA component results are available, an answer *a* is provided;

[5]The adoption of a Likert scale made it possible to compute the average and standard deviations of the user comments with respect to each answer among the top five returned by the system. It was therefore possible to replace the binary measurement of perceived usefulness, relatedness and sensitivity used in (Quarteroni and Manandhar, 2007b) in terms of total number of users with a more fine-grained one in terms of average computed over the users.

5. The system enquires whether the user is interested in submitting new queries;

6. Whenever the user wants to terminate the interaction, a final greeting is exchanged.

4.1 Choosing a Dialogue Manager

Among traditional methods for implementing information-seeking dialogue management, Finite-State (FS) approaches are the simplest. Here, the dialogue manager is represented as a Finite-State machine, where each state models a separate phase of the conversation, and each dialogue move encodes a transition to a subsequent state (Sutton, 1998). However, an issue with FS models is that they allow very limited freedom in the range of user utterances: since each dialogue move must be pre-encoded in the models, there is a scalability issue when addressing open domain dialogue.

On the other hand, we believe that other dialogue approaches such as the Information State (IS) (Larsson et al., 2000) are primarily suited to applications requiring a planning component such as closed-domain dialogue systems and to a lesser extent to open-domain QA.

As an alternative approach, we studied conversational agents ("chatbots") based on AIML (Artificial Intelligence Markup Language), such as ALICE[6]. Chatbots are based on the pattern matching technique, which consists in matching the last user utterance against a range of dialogue patterns known to the system. A coherent answer is created by following a range of "template" responses associated with such patterns.

As its primary application is small-talk, chatbot dialogue appears more natural than in FS and IS systems. Moreover, since chatbots support a limited notion of context, they can handle follow-up recognition and other dialogue phenomena not easily covered using standard FS models.

4.2 A Wizard-of-Oz Experiment

To assess the utility of a chatbot-based dialogue manager in an open-domain QA application, we conducted an exploratory Wizard of Oz experiment.

Wizard-of-Oz (WOz) experiments are usually deployed for natural language systems to obtain initial data when a full-fledged prototype is not yet available (Dahlbaeck et al., 1993) and consist in "hiding" a human operator behind a computer interface to simulate a conversation with the user, who believes to be interacting with a fully automated prototype.

We designed six tasks reflecting the intended typical usage of the system (e.g.: "Find out who painted Guernica and ask the system for more information about the artist") to be carried out by 7 users by interacting with an instant messaging platform, which they were told to be the system interface.

[6] www.alicebot.org/

The role of the Wizard was to simulate a limited range of utterances and conversational situations handled by a chatbot.

User feedback was collected mainly by using a post-hoc questionnaire inspired by the experiment in (Munteanu and Boldea, 2000), which consists of questions Q_1 to Q_6 in Table 2, col. 1, to be answered using a scale from 1="Not at all" to 5="Yes, absolutely".

From the WOz results, reported in Table 2, col. "WOz", users appear to be generally very satisfied with the system's performances: Q_6 obtained an average of $4.5\pm.5$. None of the users had difficulties in reformulating their questions when this was requested: Q_4 obtained $3.8\pm.5$. For the remaining questions, satisfaction levels were high: users generally thought that the system understood their information needs (Q_2 obtained 4) and were able to obtain such information (Q_1 obtained $4.3\pm.5$).

The dialogue manager and interface of YourQA were implemented based on the dialogue scenario and the successful outcome of the WOz experiment.

4.3 Dialogue Management Algorithms

As chatbot dialogue follows a pattern-matching approach, it is not constrained by a notion of "state": when a user utterance is issued, the chatbot's strategy is to look for a pattern matching it and fire the corresponding template response. Our main focus of attention in terms of dialogue manager design was therefore directed to the dialogue tasks invoking external resources, such as handling follow-up questions, and tasks involving the QA component.

4.3.1 Handling follow-up questions

For the *detection* of follow-up questions, the algorithm in (De Boni and Manandhar, 2005) is used, which uses features such as the presence of pronouns and the absence of verbs in the current question and word repetitions with the n previous questions to determine whether q_i is a follow-up question with respect to the current context. If the question q is not identified as a follow-up question, it is submitted to the QA component. Otherwise, the *reference resolution* strategy below is applied on q, drawing on the stack S of previous user questions:

1. If q is *elliptic* (i.e. contains no verbs), its keywords are completed with the keywords extracted by the QA component from the previous question in S for which there exists an answer. The completed query is submitted to the QA component;

2. If q contains *pronoun/adjective anaphora*, a chunker is used to find the most recent compatible antecedent in S. This must be a NP compatible in number with the referent.

3. If q contains *NP anaphora*, the first NP in S containing all the words in the referent is used to replace the latter in q. When no antecedent can be

found, a clarification request is issued by the system until a resolved query can be submitted to the QA component.

When the QA process is terminated, a message directing the user to the HTML answer frame (see Figure 1) is returned and a follow-up proposal or an enquiry about user satisfaction is optionally issued.

4.4 Implementation

To implement the dialogue manager and allow a seamless integration with our Java-based QA system, we extended the Java-based AIML interpreter Chatterbean[7]. We started by augmenting the default AIML tag set (including tags such as **<srai>** and **<that>**) with two tags: **<query>**, to seamlessly invoke the core QA module, and **<clarify>**, to support follow-up detection and resolution.

Moreover, the interpreter allows to instantiate and update a set of variables, represented as context properties. Among others, we defined:
a) **userID**, which is matched against a list of known user IDs to select a UM profile for answer extraction (see Section 5);
b) the current **query**, which is used to dynamically update the stack of recent user questions used by the clarification request detection module to perform reference resolution;
c) the **topic** of conversation, i.e. the keywords of the last question issued by the user which received an answer. The latter is used to clarify elliptic questions, by augmenting the current query keywords with those in the topic when ellipsis is detected.

Figure 1 illustrates YourQA's interactive version, which is accessible from the Web. As in a normal chat application, users write in a text field and the current session history as well as the interlocutor replies are visualized in a text area.

4.5 Interactive QA evaluation

For the evaluation of interactivity, we built on our previous results from a Wizard-of-Oz experiment and an initial evaluation conducted on a limited set of handcrafted questions (Quarteroni and Manandhar, 2007a). We chose 9 question series from the TREC-QA 2007 campaign[8]. Three questions were retained per series to make each evaluation balanced. For instance, the three following questions were used to form one task: 266.1: "When was Rafik Hariri born?", 266.2: "To what religion did he belong (including sect)?" and 266.4: "At what time in the day was he assassinated?".

Twelve users were invited to find answers to the questions to one of them by using the standard version of the system and to the second by using the interactive version. Each series was evaluated at least once using both versions of the system. At the end of the experiment, users had to give feedback about both versions of the system by filling in the satisfaction questionnaire reported in Table 2.

Table 2: Interactive QA evaluation results obtained for the WOz, Standard and Interactive versions of YourQA. Average ± st. dev. are reported.

	Question	WOz	Stand	Interact
Q_1	Did you get all the information you wanted using YourQA?	4.3±.5	4.1±1	4.3±.7
Q_2	Do you think YourQA understood what you asked?	4.0	3.4±1.3	3.8±1.1
Q_3	How easy was it to obtain the information you wanted?	4.0±.8	3.9±1.1	3.7±1
Q_4	Was it difficult to reformulate your questions when requested?	3.8±.5	-	3.9±.6
Q_5	Do you think you would use YourQA again?	4.1±.6	3.3±1.6	3.1±1.4
Q_6	Overall, are you satisfied with YourQA?	4.5±.5	3.7±1.2	3.8±1.2
Q_7	Was the pace of interaction with YourQA appropriate?	-	3.2±1.2	3.3±1.2
Q_8	How often was YourQA sluggish in replying to you?	-	2.7±1.1	2.5±1.2
Q_9	Which interface did you prefer?	-	41.7%	58.3%

Although the paired t-test conducted to compare questionnaire replies to the standard and interactive versions did not register statistical significance, we believe that the evidence we collected suggests a few interesting interpretations.

First, a good overall satisfaction appears with both versions of the system (Q_6), with a slight difference in favor of the interactive version. The two versions of the system seem to offer different advantages: while the ease of use of the standard version was rated higher (Q_3), probably because the system's reformulation requests added a challenge to users used to search engine interaction, users felt they obtained more information using the interactive version (Q_1).

Concerning interaction comfort, users seemed to feel that the interactive version understood better their requests than the standard one (Q_2); they also found it easy to reformulate questions when the former asked to (Q_6). However, while the pace of interaction was judged slightly more appropriate in the interactive case (Q_7), interaction was considered faster when using the standard version (Q_4). This partly explains the fact that users seemed more ready to use again the standard version of the system (Q_5).

7 out of 12 users (58.3%) answered the "preference" question Q_9 by saying that they preferred the interactive version. The reasons given by users in their

[7] chatterbean.bitoflife.cjb.net.
[8] trec.nist.gov

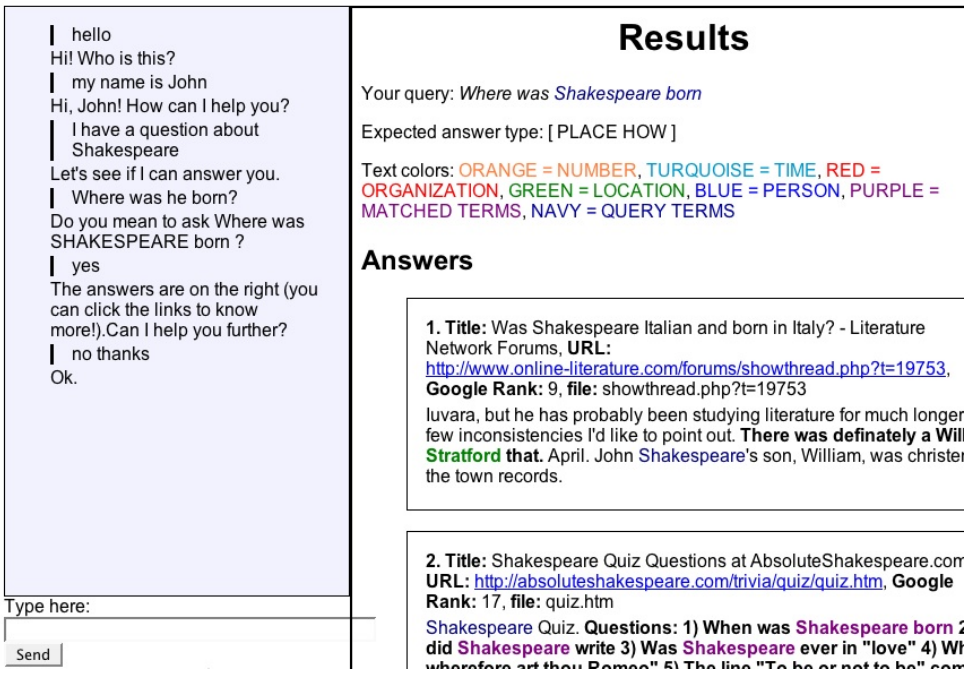

Figure 1: YourQA's interactive interface

comments were mixed: while some of them were enthusiastic about the chatbot's small-talk features, others clearly said that they felt more comfortable with a search engine-like interface. Most of the critical aspects emerging from our overall satisfactory evaluation depend on the specific system we have tested rather than on the nature of interactive QA, to which none of such results appear to be detrimental.

We believe that the search-engine-style use and interpretation of QA systems are due to the fact that QA is still a very little known technology. It is a challenge for both developers and the larger public to cooperate in designing and discovering applications that take advantage of the potentials of interactivity.

5 A Unified Model

Our research so far has demonstrated the utility of personalization and interactivity in a QA system. It is thus inevitable to regard the formulation of a unified model of personalized, interactive QA as a valuable by-product of these two technologies. In this perspective, we propose the following dialogue scenario:

1. The user interacts with the dialogue interface formulating an utterance q;

2. If q is recognized as a question, it is analyzed by the dialogue manager (DM) to detect and resolve multiple and follow-up questions;

3. As soon as a resolved version q' of q is available, the DM passes q' to the QA module; the latter processes q' and retrieves a set $Retr(q')$ of relevant documents;

4. The QA module exchanges information with the UM component which is responsible of maintaining and updating the User Model of the current user, u; Based on u, the QA module extracts a list $L(q', u)$ of personalized results from $Retr(q')$;

5. The DM produces a reply r, which is returned along with $L(q', u)$ to the user via the dialogue interface;

6. Once terminated, the current QA session is logged into the dialogue history $H(u)$, that will be used to update u;

Concerning step 4, an efficient strategy for eliciting the User Model from the user is yet to be specified at this stage: the current one relies on the definition of a context variable `userID` in the dialogue manager, which at the moment corresponds to the user's name. A number of AIML categories are created are created for YourQA to explicitly ask for the user's name, whihc is then assigned to the `userID` variable.

Figure 2 illustrates an example of a personalized, QA session in YourQA where the user's name is associated with a basic reading level UM. This affects the document retrieval phase, where only documents with simple words are retained for answer extraction.

6 Conclusions and Future Work

In this paper, we present an efficient and light-weight method to personalize the results of a Web-based QA system based on a User Model representing individual users' reading level, age range and interests. Our results show the efficiency of reading level estimation, and a

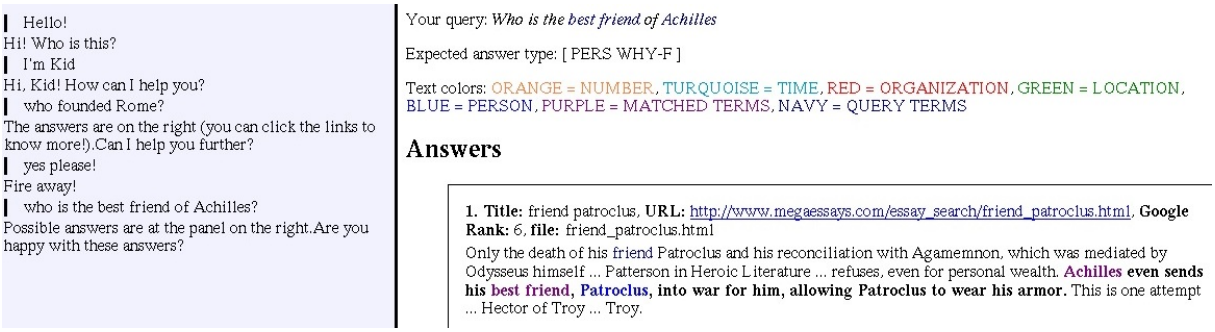

Figure 2: Screenshot from a personalized, interactive QA session. Here, the user's name ("Kid") is associated with a UM requiring a basic reading level, hence the candidate answer documents are filtered accordingly.

significant improvement in satisfaction when filtering answers based on the users' profile with respect to the baseline version of our system. Moreover, we introduce a dialogue management model for interactive QA based on a chat interface and evaluate it with optimistic conclusions.

In the future, we plan to study efficient strategies for bootstrapping User Models based on current and past conversations with the present user. Another problem to be solved is updating user interests and reading levels based on the dialogue history, in order to make the system fully adaptive.

Acknowledgements

The research reported here was mainly conducted at the Computer Science Department of the University of York, UK, under the supervision of Suresh Manandhar.

References

Belkin, N. J. and W.B. Croft. 1992. Information filtering and information retrieval: Two sides of the same coin? *Comm. ACM*, 35(12):29–38.

Collins-Thompson, K. and J. P. Callan. 2004. A language modeling approach to predicting reading difficulty. In *HLT/NAACL'04*.

Dahlbaeck, N., A. Jonsson, and L. Ahrenberg. 1993. Wizard of Oz studies: why and how. In *IUI '93*.

De Boni, M. and S. Manandhar. 2005. Implementing clarification dialogue in open-domain question answering. *JNLE*, 11.

Frank, E., G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *IJCAI '99*.

Kato, T., J. Fukumoto, F.Masui, and N. Kando. 2006. Woz simulation of interactive question answering. In *IQA'06*.

Kobsa, A. 2001. Generic user modeling systems. *UMUAI*, 11:49–63.

Kwok, C. T., O. Etzioni, and D. S. Weld. 2001. Scaling question answering to the web. In *WWW'01*.

Larsson, S., P. Ljunglöf, R. Cooper, E. Engdahl, and S. Ericsson. 2000. GoDiS—an accommodating dialogue system. In *ANLP/NAACL'00 WS on Conversational Systems*.

Magnini, B. and C. Strapparava. 2001. Improving user modelling with content-based techniques. In *UM'01*.

Maybury, M. T. 2002. Towards a question answering roadmap. Technical report, MITRE Corporation.

Moschitti, A., S. Quarteroni, R. Basili, and S. Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *ACL'07*.

Munteanu, C. and M. Boldea. 2000. Mdwoz: A wizard of oz environment for dialog systems development. In *LREC'00*.

Quarteroni, S. and S. Manandhar. 2007a. A chatbot-based interactive question answering system. In *DECALOG'07*, Rovereto, Italy.

Quarteroni, S. and S. Manandhar. 2007b. User modelling for personalized question answering. In *AI*IA'07*, Rome, Italy.

Small, S., T. Liu, N. Shimizu, and T. Strzalkowski. 2003. HITIQA: an interactive question answering system- a preliminary report. In *ACL'03 WS on Multilingual summarization and QA*.

Sutton, S. 1998. Universal speech tools: the CSLU toolkit. In *ICSLP'98*.

Teevan, J., S. T. Dumais, and E. Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *SIGIR '05*.

Voorhees, E. M. 2003. Overview of the TREC 2003 Question Answering Track. In *TREC'03*.

Witten, I. H., G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. 1999. KEA: Practical automatic keyphrase extraction. In *ACM DL*.

Creating and Querying a Domain dependent Know-How Knowledge Base of Advices and Warnings

Lionel Fontan
IRIT - UPS,
118 route de Narbonne,
31062 Toulouse Cedex, France.
antonin_follet@hotmail.fr

Patrick Saint-Dizier
IRIT - CNRS,
118 route de Narbonne,
31062 Toulouse Cedex, France.
stdizier@irit.fr

Abstract

In this paper, we present the explanation structure of procedural texts, that supports and motivates the goal-instruction structure. We focus in particular on arguments, and show how arguments of type warnings and advices can be extracted. Finally, we show how a domain dependent know-how textual knowledge base can be constructed and queried.

1 Introduction

Procedural texts consist of a sequence of instructions, designed with some accuracy in order to reach a goal (e.g. assemble a computer). Procedural texts may also include subgoals. These are most of the time realized by means of titles and subtitles. The user must carefully follow step by step the given instructions in order to reach the goal.

The main goal of our project is to analyse the structure of procedural texts in order to efficiently and accurately respond to How-to ? questions. This means identifying titles (which convey the main goals of the text), sequences of instructions serving these goals, and a number of additional structures such as prerequisites, warnings, advices, illustrations, etc.

In our perspective, procedural texts range from apparently simple cooking recipes to large maintenance manuals. They also include documents as diverse as teaching texts, medical notices, social behavior recommendations, directions for use, assembly notices, do-it-yourself notices, itinerary guides, advice texts, savoir-faire guides etc. (Aouladomar et al., 2005). Procedural texts follow a number of structural criteria, whose realization may depend on the author's writing abilities, on the target user, and on traditions associated with a given domain. Procedural texts can be regulatory, procedural, programmatory, prescriptive or injunctive. The work we report here was carried out on a development corpus of French texts taken from the Web from most of the various domains cited above.

Argument extraction is not yet a very active area, althought it has obvious uses in question answering, in decision theory, etc. For example, extracting arguments from legal texts (ICAIL 2005) or for answering opinion questions is a major challenge of primary use.

We have developed a quite detailed analysis of procedural texts, identifying their main basic components as well as their global structure. For that purpose, we have defined two levels: a segmentation level that basically tags structures considered as terminal structures (titles, instructions, advices, prerequisites, etc.) and a grammar level that binds these terminal structures to give a global structure to procedural texts (Delpech et al. 2008). This structure is textual and dedicated only to elements relevant to procedurality.

Procedural texts are complex structures, they often exhibit a quite complex rational (the instructions) and 'irrational' structure which is mainly composed of advices, conditions, preferences, evaluations, user stimulations, etc. They form what is called the explanation structure, which motivates and justifies the goal-instructions structure, which is the backbone of procedural texts. A number of these elements are forms of argumentation, they provide a strong and essential internal cohesion and coherence to procedural texts.

Coling 2008: Proceedings of the workshop on Knowledge and Reasoning for Answering Questions, pages 41–48
Manchester, August 2008

An important aspect of this project is the accurate identification of the explanation structure as found in procedural texts in order (1) to better understand explanation strategies deployed by humans in precise, concrete and operational situations and (2) to build a knowledge base of advices and warnings related to an application domain, that reflects several forms of know-how on this domain. Such repositories exist, but they have been build completely manually, by various users, often in a wiki fashion. Our goal is then to allow users not only to query procedural texts via How to questions, but also to create and to access to a repository of advices and warnings (basically Why questions and some How-to questions to a lesser extent) about a certain task.

We have already studied the instructional aspects of procedural texts and implemented a quite efficient prototype within the TextCoop project (Delpech et al. 2008) that tags text with dedicated XML tags. In this paper, after a brief categorization of explanation structure as found in our corpus of procedural texts, we focus on the argumentation structure via the recognition of warnings and advices. Then, we show how a textual knowledge base of advices and warnings can be produced and how it can be queried.

2 The explanation structure in procedural texts

We first present, in this section, the general organization of the explanation structure as it emerged from corpus analysis. Then we develop the major component of procedural texts: the instructional compound.

2.1 A global view of the explanation structure

From our development corpus, we established a classification of the different forms explanations may take. Basically, the explanation structure is meant to guide the user by making sure that he will effectively realize actions as they are specified, via e.g. threats, rewards, evaluations, advices and warnings. The main structures are facilitation and argumentation structures; they are either global (they are adjoined to goals, and have scope over the whole procedure) or local, included into instructional compounds, with a scope local to the instructional compound. This latter case is by far the most frequently encountered. These structures are summarized as follows (the terms we use are either borrowed from works on rhetorical relations or are just ours if none exist):

- **facilitation structures**, which are rhetorical in essence (Kosseim et al 2000) (Van der Linden 1993), correspond to *How to do X ?* questions, these include two subcategories:
  (1) user help, with: hints, evaluations and encouragements and
  (2) controls on instruction realization, with two cases: (2.1) controls on actions: guidance, focusing, expected result and elaboration and (2.2) controls on user interpretations: definitions, reformulations, illustrations and also elaborations.

- **argumentation structures**, corresponding to *why do X ?* questions.

  These have either:
  (1) a positive orientation with the author involvement (promises) or not (advices and justifications) or
  (2) a negative orientation with the author involvement (threats) or not (warnings).

In what follows, we will mainly concentrate on this second point, and in particular on warnings and advices which are the most frequently encountered (since there are rarely involvements from the author). These will be used to construct the know-how knowledge base. Argumentation structures are relatively general to an applications domain, while facilitation structures are much more specific to the text and the targeted audiences. There are several ways of defining and approaching argumentation. Without entering any debate, we consider here the approach where an argument is composed of one or more supports associated with a certain statement, as in the following warning: *carefully plug in your mother card vertically, otherwise you will most likely damage its connectors*. where if the intruction (carefully plug in...) is not correctly realized, the user know the consequences.

2.2 From instructions to instructional compounds

In most types of texts, we do not find just sequences of simple instructions but much more

complex compounds composed of clusters of instructions, that exhibit a number of semantic dependencies between each other, that we call **instructional compounds**. These are organized around a few main instructions, to which a number of subordinate instructions, warnings, arguments, and explanations of various sorts may possibly be adjoined. All these elements are, in fact, essential in a compound for a good understanding of the procedure at stake.

An instructional compound has a relatively well organized discourse structure, composed of several layers, which are:

- The **goal and justification** level, which has wider scope over the remainder of the compound, indicates motivations for doing actions that follow in the compound (e.g. *in your bedroom, you must clean regularly the curtains...*, which here motivates actions to undertake). It gives the fundamental motivation of the compound.

- The **instruction kernel structure**, which contains the main instructions. These can be organized temporally or be just sets of actions. Actions are identified most frequently via the presence of action verbs (in relation to the domain) in the imperative form, or in the infinitive form introduced by a modal. We observed also a number of subordinated instructions forms adjoined to the main instructions. These are in general organized within the compound by means of rhetorical relations, introduced below.

- The **deontic and illocutionary force structures**: consist of marks that operate over instructions, outlining different parameters. These linguistic structures play a major role in argumentation:
  - deontic: obligatory, optional, forbidden or impossible, alternates (or),
  - illocutionary and related aspects: stresses on actions: necessary, advised, recommended, to be avoided, etc. These marks are crucial to identify the weight of an argument.

- a **temporal structure** that organizes sequences of instructions (and, at a higher level, instructional compounds). In general, the temporal structure is very simple, with sequences of actions to carry out. In some cases, parallel actions are specified, which partially overlap.

- The **conditional structure**: introduces conditions over instructions within the compound or even over the whole instructional compound. We encounter quite a lot of structures organizing mutually exclusive cases.

- the **causal structure** that indicates the goal of an action. We identify four types of causal relations, following (Talmy 2001): intend-to (direct objective of an action: *push the button to start the engine*), Instrumented (*use a 2 inch key to dismount the door*), Facilitation (*enlarge the hole to better empty the tank*) and Continue (*keep the liquid warm till its colour changes*).

- The **rhetorical structure** whose goal is to enrich the kernel structure by means of a number of subordinated aspects (realized as propositions, possibly instructions) among which, most notably: enablement, motivation, circumstance, elaboration, instrument, precaution, manner. A group of relations of particular interest in this paper are arguments, developed hereafter.

Explanations and arguments help the user understand why an instruction must be realized and what are the risks or the drawbacks if he does not do it properly. An example of an instructional compound is:

[$_{instructional\ compound}$
[$_{Goal}$ To clean leather armchairs,]
[$_{argument:advice}$
[$_{instruction}$ choose specialized products dedicated to furniture,
[$_{instruction}$ and prefer them colourless]],
[$_{support}$ they will play a protection role, add beauty, and repair some small damages.]]]

We have here an argument of type advice which is composed of 2 instructions (later called a conclusion) and a conjunction of three supports which motivate the 2 instructions.

3 Identifying arguments in procedures

In this section let us first give a quite informal definition of what an argument is, and how it interacts with the goal-instructions structure. Let us then focus on warnings and advices which are, by far, the most frequently encountered structures. Most warnings and advices are included into instructional compounds.

3.1 Argumentation and Action theories

Roughly, argumentation is a process that allows speakers to construct statements for or against another statement called the conclusion. These former statements are called supports. The general form of an argument is : **Conclusion 'because' Support** (noted as $C\ because\ S$). In natural language, conclusions often appear before the support, but they may also appear after. A conclusion may receive several supports, possibly of different natures (advices and warnings). Arguments may be more or less strong, they bear in general a certain weight, induced from the words they contain (Anscombre et al. 1981), (Moeschler 1985), (Amgoud et ali. 2001). In natural contexts, this weight is somewhat vague, and only general classes can be produced, e.g. from light to strong.

In the case of procedural texts, the representation and the role of arguments in a text can be modelled roughly as follows. Let G be a goal which can be reached by the sequence of instructions $A_i,\ i\ \in\ [1,n]$, whatever their exact temporal structure is. A subset of those instructions is interpreted as arguments where each instruction (A_j, viewed as a conclusion) is paired with a support S_j that stresses the importance of A_j (*Carefully plug in your mother card vertically, otherwise you will damage the connectors*). Their general form is: $A_j\ because\ S_j$ (we use here the term 'because' which is more vague than the implication symbol used in formal argumentation, because natural language is not so radical). Supports S_k which are negatively oriented are warnings whereas those which are positively oriented are advices. Neutral supports simply introduce basic explanations.

Similarly to the principles of argument theory, but within the framework of action theory (e.g. Davidson 2003), if A_j is associated with a support of type warning S_j then if A_j is not realized correctly, the warning S_j is 'active' and attacks the goal G, i.e. it makes its realization more difficult, if not impossible. Conversely, if S_j is an advice, it supports the goal G, making its full realization easier, or providing better results if A_j is executed. Note however that there is an implicit gradability in the realization of an action, which may be more or less accurately and completely realized. In that case, negative or positive consequences on the main goal evolve accordingly.

Supports can themselves receive supports : *don't add natural fertilizer, this may attract insects, which will damage your young plants*. In the same range of ideas, instructions A_j which are advices or warnings have a different status than 'normal', unsupported instructions (although one can say that most of them could be associated with an implicit support such as *otherwise you will fail*). Advices are often optional instructions: they are a kind of invitation to do the associated action for better results, whereas warnings are an incitation to be more careful. Therefore, instructions in a procedure do not have all the same operational strength and status.

As can be noted, our definition includes terms which are gradual: 'more difficult', 'easier', because in practice, failing to realize an instruction properly does not necessarily means that the goal cannot be reached, but the user will just be less successful, for various reasons. In the natural language expressions of conclusions (the A_j) as well as of supports, there are many modals or classes of verbs (like risk verbs) that modulate the consequences on G, contrast for example:
use professional products to clean your leathers, they will give them a brighter aspect. with:
carefully plug in your mother card vertically, otherwise you will most likely damage its connectors.
In the latter case, the goal 'mounting your own PC' is likely to fail, whereas in the former, the goal 'cleaning your leathers' will just be less successful.

3.2 Processing arguments

From the above observations, we have defined a set of patterns that recognize instructions which are conclusions and their related supports. We defined those patterns from a development corpus of about 1700 texts from various domains (cooking, do it yourself, gardening, video games, social advices, etc.). The study is made on French, English glosses are given here for ease of read-

ing. The recognition problem is twofold: identifying propositions as conclusions or supports by means of specific linguistic marks (sometimes we also found a few typographic marks), and then delimiting these elements. In general, boundaries are either sentences or, by default, instructional compound boundaries. In procedural texts, roughly, the proportion of advices and warnings is almost equivalent.

3.2.1 Processing warnings

Warnings are basically organized around a unique structure composed of an 'avoid expression' combined with a proposition. The variations around the 'avoid expressions' capture the illocutionary force of the argument via several devices, ordered here by increasing force :
(1) 'prevention verbs like avoid' NP / to VP (*avoid hot water*)
(2) do not / never / ... VP(infinitive) ... (*never put this cloth in the sun*)
(3) it is essential, vital, ... to never VP(infinitive).
In cases where the conclusion is relatively weak in terms of consequences, it may not have any specific mark, its recognition is then based on the observation that it is the instruction that immediately precedes an already identified support.

Supports are propositions which are identified from various marks:
(1) via connectors such as: *sinon, car, sous peine de, au risque de* (otherwise, under the risk of), etc. or via verbs expressing consequence,
(2) via negative expressions of the form: *in order not to, in order to avoid, etc.*
(3) via specific verbs such as risk verbs introducing an event (*you risk to break*). In general the embedded verb has a negative polarity.
(4) via the presence of very negative terms, such as: nouns: *death, disease, etc.*, adjectives, and some verbs and adverbs. We have a lexicon of about 200 negative terms found in our corpora.

Some supports have a more neutral formulation: they may be a portion of a sentence where a conclusion has been identified. For example, a proposition in the future tense or conditional following a conclusion is identified as a support. However, as will be seen below, some supports may be empty, because they can easily be inferred by the reader. In that case, the argument is said to be truncated.

Patterns are implemented in Perl and are included into the TextCoop software. From the above observations, with some generalizations and the construction of lexicons of marks, we have summarized the extraction process in only 8 patterns for supports and 3 patterns for conclusions. Pattrens are basically morpho-lexical, with the need to recognize a few local structures, treated by means of local automata. A pattern in Perl has the following form:

```
 (PRO:PER--Modalite +)?--
evit(ez|er)--(\w+ )*--##
```

with modalite = devoir, veiller a, etre essentiel, etc. Some local automata are associated with most patterns in order to make them as generic as possible. In our programme, Perl scripts are treated one after the other, in sequence. We do not have any efficieny requirement since these treatments are realized in batch mode. However, for the whole processing, we tag abouit 200 Mo of text per hour on a standard 3GhZ Pentium machine.

3.2.2 Evaluation

In procedural texts, arguments are tagged by XML tags. We carried out an indicative evaluation (e.g. to get improvement directions) on a corpus of 66 texts over various domains, containing 302 arguments, including 140 advices and 162 warnings. This test corpus was collected from a large collection of texts from our study corpus. Domains are in 2 categories: cooking, gardening and do it yourself, which are very prototypical, and 2 other domains, far less stable: social recommendations and video games solutions. Arguments were manually tagged in these texts, and a comparison was made with the output of the system. Therefore, we report below the recall, the precision being almost 100% (very little noise).

We get the following results for warnings:

conclusion recognition	support recognition	(3)	(4)
88%	91%	95%	95%

(3) conclusions well delimited (4) supports well delimited, with respect to warnings correctly identified.

As far as warnings are concerned, results are really good. Errors are very diverse, some of them involve uses of the verb *pouvoir* (to be able to) and the auxiliary *être* (to be).

3.2.3 Processing Advices

Conclusions of type advice are identified essentially by means of two types of patterns (in French):
(1) advice or preference expressions followed by an instruction. The expressions may be a verb or a more complex expression: *is advised to, prefer, it is better, preferable to, etc.*,
(2) expression of optionality or of preference followed by an instruction: *our suggestions: ...*, or expression of optionality within the instruction (*use preferably a sharp knife*).
In addition, as for warnings, any instruction preceding a support of type advice is a conclusion.

The first pattern above is recogned by the following script:

```
 ceci|cela|NOM|PRO:
PER+--tre?--ADV?--Verb/
advice exporession--(\w+ )*--##
```

Supports of type advice are identified on the basis of 3 distinct types of patterns:
(1) Goal exp + (adverb) + positively oriented term. Goal expressions are e.g.: in order to, for, whereas adverb includes: better (in French: mieux, plus, davantage), and positively oriented term includes: nouns (savings, perfection, gain, etc.), adjectives (efficient, easy, useful, etc.), or adverbs (well, simply, etc.). For this latter class of positively oriented terms we constructed a lexicon that contains about 50 terms. terms.
(2) goal expression with a positive consequence verb (favour, encourage, save, etc.), or a facilitation verb (improve, optimize, facilitate, embellish, help, contribute, etc.),
(3) the goal expression in (1) and (2) above can be replaced by the verb 'to be' in the future: *it will be easier to locate your keys*.

Similarly as above, we carried out an indicative evaluation on the same corpus as above, with the same experimental conditions. We get the following results for advices:

conclusion recognition	support recognition	(3)	(4)	(5)
79%	84%	92%	91%	91%

(3) conclusions well delimited, (4) supports well delimited, both with respect to advices correctly identified. (5) support and conclusion correctly related.

A short example of an annotated text is given in Fig. 1 below.

4 Constructing and Querying a know-how textual database

Besides studying the textual structure of procedural texts and responding to How-to questions (Delpech et al. 2007) from the analysis of these texts, a major application of this work is the construction of **domain know-how knowledge base**, which is probably quite basic, but which could be subject to interesting generalizations. Obviously, to make this knowledge optimal, it would be useful to associate with every statement a formal representation that supports inference, data fusion, etc.

This domain know-how knowledge base of advices, hints and warnings is of much importance for different types of users who have a procedure to realize a task but who want to know more before starting. Some psychological experiments have in fact shown that, besides instructions given in procedural texts, users are very much interested in what remains implicit in those texts: what you are supposed to know or care about (but have no means to ask). This know-how textual database is aimed to fill in this kind of gap.

The work presented hereafter is still exploratory, since the task is quite complex. The domain know-how textual database is planned to be either directly consulted by users, or queried by means of requests in natural language or keywords.

4.1 Constructing a text database of domain know-how

There are repositories of advices organized by sector of activity available on the Web (e.g. http://www.conseils-gratuit.com). These are realized manually: most of these advices come from hints sent by readers of these pages. These repositories contain in general simple advices and also small procedures which are hints to better realize a certain task.

In our approach, the text units that we have access to are either (1) procedural texts decomposed into subgoals when they are large (e.g. the different phases of assembling a computer), or (2) instructional compounds. Compounds roughly correspond to the various advice forms found in man-

[$procedure$
[$title$ How to embellish your balcony
[$Prerequisites$ 1 lattice, window boxes, etc.]
....
[$instructional-compound$ In order to train a plant to grow up a wall, select first a sunny area, clean the floor and make sure it is flat.....
[$Argument$ [$Conclusion$:$Advice$ You should better let a 10 cm interval between the wall and the lattice.]
[$Support$:$Advice$ This space will allow the air to move around, which is beneficial for the health of your plant.]
.....]]]]

Figure 1: An annotated procedure

ually realized repositories of advices. Advices and warnings mainly appear within these instructional compounds. However, compounds being inserted into a larger procedure may be somewhat elliptical in some cases. Therefore, the textual database we are constructing will contain titles (to settle context) and compounds.

Let us now present the construction of the domain know-how textual database of advices and warnings. At this stage, this is an experimental tentative that needs further improvements and evaluation. We first process texts by domain, according to our corpus (about 8000 texts). The main functions of this processing are:
(1) cleaning web pages from irrelevant data (adds, forums, summaries, links, etc.),
(2) XML tagging the instructional aspects, with dedicated tags: tagging titles (and reconstructing the numerous titles which are incomplete, with missing verb or object, and tagging instructional compounds and prerequisites, and
(3) tagging within instructional compounds advices and warnings based on the patterns given above.

In the textual database, the first level of structure is domains: `house, coocking, administration, health, garden, computer, do it yourself, animals, beauty, society`.

Next, below each of these domain top nodes, we have a list of items that correspond to procedures main titles (e.g. *boucher un trou avec du platre (fill up a hole with plaster)*. Since, for most domains we have several hundreds of documents, we need to organize those titles and abstract over them. This is being organized around two axis:
(1) task oriented: where action verbs are grouped on the basis of closely related terms to form a single title (for that purpose we use our verb lexical base (Volem)). A second level of generalization is carried out by skipping adjuncts, therefore we have items like: 'repairing walls' independently of the material or the technique used, e.g. with plaster. mastic, cement.
(2) object oriented: where we only keep track of the objects, viewed as a theme: wall, wood, plaster, etc. so that the user can access the different operations these objects may undergo.
These revised titles form a second level in the structure of the know-how textual knowledge base.

Below these links, we have the list of relevant web pages. Each of these pages is associated with an index composed of the set of titles it contains and the list of supports identified (reconstructed supports are not yet included). Titles are used to make the procedure context more precise so that the scope of supports is more clear, since some supports are vague. A short example is given in Fig. 2 below. Supports which are too vague to be of any use are filtered out. At the moment we are studying various forms of filters based on the type of words they contain and their relevance.

4.2 Querying the know-how textual database

In general, attempting to match queries directly with supports in order to get the advice, i.e. the associated conclusion does not lead to the best results because supports are often incomplete or they contain a lot of pronominal references. Our matching procedure therefore includes the taking into account of the page title, or subtitles together with support contents. It seems that this leads to better results in terms of accuracy and relevance.

Related to Fig. 2, a query could be: *how to get smooth plaster surfaces on a wall ?*. There is no procedural text that corresponds to this query,

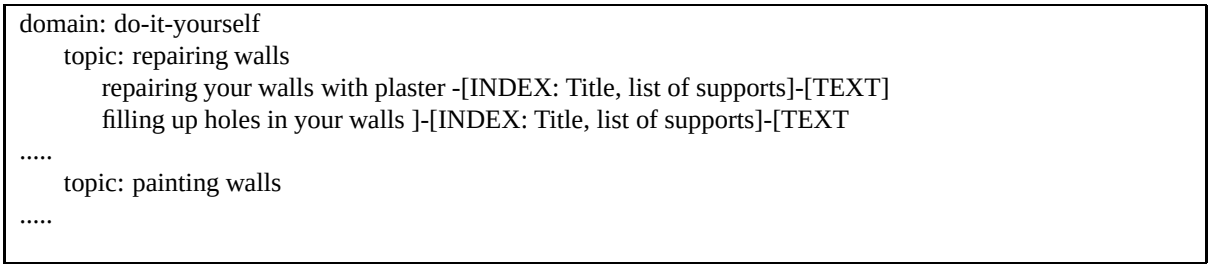

Figure 2: A text database index

which is rather an advice request. Answering this question is realized by the following steps:
(1) based on keywords which appear as objects in the query, select a domain and a topic in the knowledge base.
(2) then, over the topics selected, match the query with one or more supports. Matching is obviously not direct and requires, as in most systems, some flexibility. Of interest here are adjectives, which abound in this type of question, for which we need to develop scales that capture the different language expressions of the properties they characterize. This type of scale, in (Cruse 1986), is called non branching proportional series. For example 'smooth' will appear on a scale of type 'surface granularity' that includes other adjectives such as rough, grainy, etc.

5 Perspectives

The work presented here complements the tagging of titles and instructional compounds in procedural texts of various domains, as reported in (Delpech et al. 2008). We analyzed the forms arguments of type advice and warning may take, and have implemented and tested a system that tags those structures and attempts at reconstructing empty supports. At this level, there is still linguistic and formal work to be carried out, for example to evaluate the illocutionary force of arguments and to better settle this work within action theory. We believe we have a very useful corpus of examples of arguments, of much interest for research in argumentation theory.

In a second stage, we have now established a first version of criteria to construct from these arguments a domain know-how textual database, that users can query to get additional information when realizing a task, often information which remains implicit in a procedure, but that users do need to operate safely and efficiently. The construction of such a repository is a complex task that we will pursue, together with an analysis of how it can be queried accurately.

Credits We thank the French ANR-RNTL research programme for supporting this project. We also thank very warmly Leila Amgoud for discussions on argumentation, Daniel Kayser for comments on this paper, and 3 anonymous reviewers.

References

Amgoud, L., Parsons, S., Maudet, N., 2001, *Arguments, Dialogue, and Negotiation*, in: 14th European Conference on Artificial Intelligence, Berlin.

Anscombre, J.-Cl. Ducrot, O., 1981, *Interrogation et Argumentation*, in Langue francaise, no 52, L'interrogation, 5 - 22.

Aouladomar, F., Saint-dizier, P., 2005, *Towards Answering Procedural Questions*, Workshop KRAQ05, IJCAI05, Edinburgh.

Cruse, A., 1986, *Lexical Semantics*, Cambridge Univ. Press.

Davidson, D., 1963, *Actions, Reasons, and Causes*, Journal of Philosophy, 60.

Delpech, E., Saint-Dizier, P., 2008, Investigating the Structure of Procedural Texts for Answering How-to Questions, LREC 2008, Marrakech.

Kosseim, L., Lapalme, G., 2000, *Choosing Rhetorical Structures to Plan Instructional Texts*, Computational Intelligence, Blackwell, Boston.

Moschler, J., 1985, *Argumentation et Conversation, Eléments pour une Analyse Pragmatique du Discours*, Hatier - Crédif.

ICAIL, 2005, Automatic semantics extraction in law documents, proceedings, C. Biagili et ali. (ed), ACM ICAIL publications, Stanford.

Vander Linden, K., 1993, *Speaking of Actions Choosing Rhetorical Status and Grammatical Form in Instructional Text Generation* Thesis, University of Colorado.

Author Index