Question Classification by Structure Induction



Menno van Zaanen **Luiz Augusto Pizzato** **Diego Mollá**

Centre for Language Technology
Macquarie University
Sydney, Australia
{menno,pizzato,diego}@ics.mq.edu.au

Abstract

*We introduce a new approach (and several implementations) to the task of **question classification**. The approach **extracts structural information using machine learning techniques and the patterns found are used to classify the questions**. This approach fits in between the machine learning and handcrafting of regular expressions (as it was done in the past) and combines the best of both: classifiers can be generated automatically and the output can be investigated and manually optimised if needed.*

Question Classification

Before a question answering (QA) system can answer a question, it needs to have an idea what the question is about. One of the principal tasks of the question analysis stage of a QA system is the determination of the expected answer type (EAT). Finding the EAT of a question is called question classification (or EAT classification) [Hermjakob 2001].

Our approach combines the two main methods to question classification: machine learning and pattern matching. Using machine learning, patterns are extracted from the training data. These patterns serve as regular expressions during the classification task.

Alignment-Based Learning

Alignment-Based Learning (**ABL**) is a **generic unsupervised grammatical inference framework that learns structure from plain text sentences**. The underlying idea of ABL is that constituents can be interchanged. To give an example, if we exchange the noun phrase *the man* in the sentence *He sees the man* with another noun phrase *a woman*, we get another valid sentence: *He sees a woman*. This process can be reversed (by aligning sentences) and possible constituents, called hypotheses can be found. An example of a toy corpus with the learned structures marked with parenthesis is:

Q Type	Structure
"DESC"	*(What) (is (caffeine))*
"DESC"	*(What) (is (Teflon))*
"LOC"	*(Where) is (Milan)*
"LOC"	*What (are the twin cities)*

The ABL Classifier

The structures learned can be used to build regular expressions. We experimented with two methods:

- **hypo**: The words in the hypotheses (within brackets) are selected.
- **unhypo**: The words that are left after removing the hypotheses are selected.

Q Type	hypo	unhypo
"DESC"	*/what/ /is caffeine/ /caffeine/*	*/is caffeine/ /what/ /what is/*
"DESC"	*/what/ /is teflon/ /teflon/*	*/is Teflon/ /what/ /what is/*
"LOC"	*/where/ /milan/*	*/is Milan/ /where is/*
"LOC"	*/are the twin cities/*	*/What/*

The candidate question is tested on all the regular expressions and is assigned the EAT with the highest count. We experimented with two variations:

- **default**: Sum the frequencies stored in each regular expression.
- **prior**: Count the EATs.

E.g. *What is penicillin* (unhypo learning method):

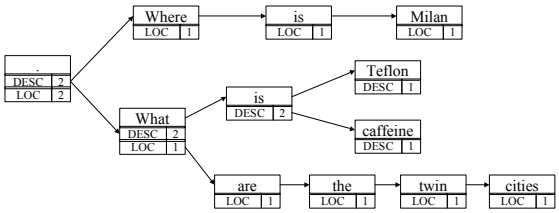
default	prior
/what/ : 2 DESC, 1 LOC */what is/* : 2 DESC	*/what/* : 1 DESC, 1 LOC */what is/* : 1 DESC
Total: 4 DESC, 1 LOC	Total: 2 DESC, 1 LOC
EAT: DESC	EAT: DESC

Trie Structures

A word-based trie structure groups sentences by their word prefixes. Thus, a trie T(S) is defined by the recursive rule:

$T(S) = \{T(S/a_1), T(S/a_2), \ldots T(S/a_r)\}$

S/a_n is the set of strings that contains all strings of S that start with the word an but stripped of that initial element [Clement et al. 1998].



The Trie Classifier

During the learning phase, each node in the trie contains the EAT and frequency information.

Classification is trivial if the question is part of the training set. Classification for unseen questions is done by **skipping non-matching nodes** in a methodical way: if Q = $\{q_1, q_2, \ldots, q_i, \ldots, q_n\}$ has found a matching node up to word q_i but q_{i+1} does not match, then:

1. Find all possible continuations by skipping one node (say β) in the trie.
2. Select the continuation with the highest frequency.

We experimented with two methods:

- **strict**: β and q_{i+1} must have the same PoS tag.
- **flex**: β and q_{i+1} must have the same PoS tag. If there are no candidates, consider the next node of the subtrie and try again, until a node in the trie has the same PoS tag as q_{i+1}.

Results

We used the question categories and training data from [Zhang and Sun Lee, 2003]:

- 5,452 training questions and 500 test questions.
- 6 coarse-grained classes and 50 fine-grained classes (we used the coarse-grained classes only).

The baseline selects the most frequent class only.

			words	PoS
Baseline			0.188	0.188
ABL	hypo	default	0.516	0.682
		prior	0.554	0.624
	unhypo	default	0.652	0.638
		prior	0.580	0.594
Trie	strict		0.844	0.812
	flex		0.850	0.794

For comparison, the results reported by [Zhang and Sun Lee, 2003] for bag-of-words and gab-of-ngrams versions ranged between 75.6% and 87.4%

References

[Clement et al., 1998] J. Clément, P. Flajolet, and B. Vallée. The analysis of hybrid trie structures. In Proc. ACMSIAM Symposium on Discrete Algorithms, pages 531–539, Philadelphia:PA, USA, 1998. SIAM Press.

[Hermjakob, 2001] Ulf Hermjakob. Parsing and question classification for question answering. In Proc. ACL/EACL Workshop on Open-Domain Question Answering, 2001.

[Zhang and Sun Lee, 2003] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In Charles Clarke, Gordon Cormack, Jamie Callan, D avid Hawking, and Alan Smeaton, editors, Proc. SIGIR, pages 26–32, New York:NY, US, 2003. ACM Press.

This work is supported by the Australian Research Council, ARC Discovery Grant no. DP0450750.