Automated Focus Extraction for Question Answering over Topic Maps

Rani Pinchuk[1], Tiphaine Dalmas[2], and Alexander Mikhailian[1]

[1] Space Applications Services, Leuvensesteenweg 325, B-1932 Zaventem, Belgium
{rani.pinchuk,alexander.mikhailian}@spaceapplications.com
[2] Aethys
tiphaine.dalmas@aethys.com

Abstract. This paper describes the first stage of question analysis in Question Answering over Topic Maps. It introduces the concepts of asking point and expected answer type as variations of the question focus. We identify the question focus in questions asked to a Question Answering system over Topic Maps. We use known machine learning techniques for expected answer type extraction and implement a novel approach to the asking point extraction. We also provide a mathematical model to predict the performance of the system.

1 Introduction

Question Answering (QA) is the task concerned with the delivery of a concise answer to a question expressed in natural language. An answer is extracted or constructed from textual data - whether (semi-)structured or structured. As in many other tasks dealing with natural language understanding, a severe bottleneck in QA lies in the design of a generic or at least adaptive technology that could be re-used to extract information from different data sources and in different domains, not to mention in multiple languages.

This work addresses domain portable QA over Topic Maps (TM). That is, a QA system capable of retrieving answers to a question asked against any particular topic map. In this paper, we present test results on the Italian Opera topic map[3].

In semi-structured data, answers are often associated with annotated types (i.e. La Bohéme is of type opera). The question focus is the type of the answer in the question terminology. For example, in the question *What operas did Puccini compose?* the focus is *operas* because the answers we are looking for are operas. That is, if the focus is mapped to the type opera, then all the operas available in the source data will be possible answers.

In QA over Topic Maps, if an answer is explicitly available in the topic map, the topic map can usually provide the type of this answer. For example, for the question *Who is the composer of La Bohéme*, the answer is *Puccini* and as can be seen in Figure 1, the type of this answer is *Composer*.

[3] http://www.ontopedia.net/omnigator/models/topicmap_nontopoly.jsp?tm=ItalianOpera.ltm

Maicher, L.; Garshol, L. M. (Eds.): Linked Topic Maps. Fifth International Conference on Topic Maps Research and Applications, TMRA 2009 Leipzig, Germany, November 12–13, 2009 Revised Selected Papers. Leipziger Beiträge zur Informatik. ISBN 978-3-941608-06-1

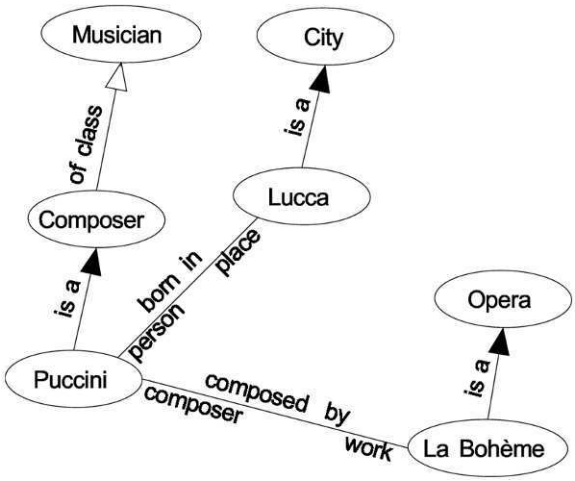

Fig. 1. A graphical representation of a topic map

In this paper, we concentrate on an empirical approach to question classification and focus extraction. The module is based on one dynamic and one static layer, contrasting with previous work that uses static taxonomies [7].

2 Focus Definition and Evaluation

Focus extraction is particularly important for QA over semi-structured data as many such data formats allow encoding of some kind of parent-child relationship, be it via the hierarchical structure of an XML document[4], OWL relationships[5] or *supertype-subtype* and *type-instance* associations in Topic Maps.

We will use the term *asking point* or AP when the type of the answer is explicit, e.g. the word *operas* in the question *What operas did Puccini write?*

We will use the term *expected answer type* or EAT when the type of the answer is implicit but can be deduced from the question using formal methods. The question *Who is Puccini?* implies that the answer is a person. That is, *person* is the expected answer type.

EATs are backed up by a taxonomy while APs can be considered dynamic types. We consider that AP takes precedence over the EAT. That is, if the AP (the explicit focus) has been successfully identified in the question, it is considered as the type of the question, and the EAT (the implicit focus) is left aside.

The following question illustrates this approach:

Who is the librettist of La Tilda?

[4] http://www.w3.org/TR/xml11
[5] http://www.w3.org/TR/2004/REC-owl-features-20040210

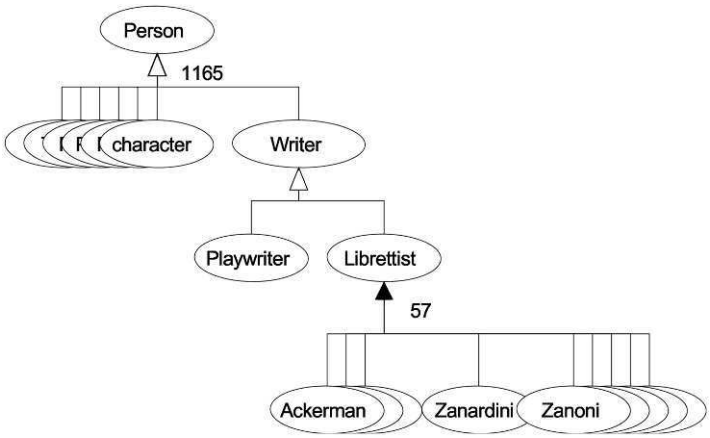

Fig. 2. Another excerpt from a topic map

The answer to this question is represented by the topic *Zanardini* and as can be seen in Figure 2, the type of this answer is *Librettist*. The AP of the question is *librettist*, while the EAT class is *HUMAN*. If we map (or "anchor") the AP to the topic map successfully, we get the type *Librettist* which has 57 instances (one of which is the actual answer to the given question). Successfully mapping the EAT class provides us with the topic *Person*, which is an ancestor topic for 1065 topics that represent people.

In order to measure the performance of AP and EAT in the example above, one approach can be to compare the type of the answer in the topic map with the anchored AP or EAT. However, the measures provided by this approach are binary (equal/not equal) and will not reflect the fact that one type might be a sub-type of another type (i.e. *Librettist* is a sub-type of *Person*).

Another approach, which is chosen here, is to measure the performance by considering the instances of the topic types. Because the answers to the questions are annotated with constructs from the topic map, the type of answer for a question can be automatically induced as it is the lowest common type of the annotated answers.

For instance, in the example above, *Librettist* is our lowest common topic type, and leads to relevant instances such as *Ackerman*, *Zanardini*, *Zanoni*, one of which is the correct answer. The extracted AP, *librettist*, anchored to the topic type *Librettist* leads to the same set of instances. On the other hand, the taxonomy class *HUMAN* is anchored to the topic type *Person* and leads to a much larger set of instances representing all the people referenced by the topic map.

Therefore, we can consider the instances of the type of the actual answer as the relevant instances, and the instances of the type anchored to the AP or the EAT as the retrieved instances. This way, the precision (P) and recall (R) for

Table 1. Precision and Recall of AP and EAT for Topic Maps

Name	Precision	Recall
AP	0.311	0.30
EAT	0.089	0.21

Table 2. Precision and Recall of AP and EAT for Topic Maps, taking into account the branching

Name	Count	Precision	Recall
AP when AP exists	55	0.565	0.546
EAT when AP exists	55	0.379	0.109
EAT when no AP	38	0.151	0.333

the example above can be calculated as follows:

$$R = \frac{|\{relevant\} \bigcap \{retrieved\}|}{|\{relevant\}|}$$

$$P = \frac{|\{relevant\} \bigcap \{retrieved\}|}{|\{retrieved\}|}$$

For the question above, we find that AP is more specific ($P_{AP} = 57/57 = 1$) than EAT ($P_{EAT} = 57/1165 = 0.049$). Both AP and EAT have 100% coverage ($R_{AP} = 57/57 = 1$ and $R_{EAT} = 57/57 = 1$).

The claim that the exploitation of AP provides indeed better performance in QA over Topic Maps has been tested with 100 questions over the Italian Opera topic map [11][6]. AP, EAT and the answers of the questions were manually annotated. The answers of the questions were annotated as topic map constructs (i.e. a topic or an occurrence).

The annotated AP and EAT were anchored to the topic map using simple anchoring methods (the details of which is out of the scope of this paper, however, these methods included exact match, matching per words, conversion the words to singular or plural, and synonyms).

Out of 100 questions, APs were annotated in 55 questions. For the remaining 45 questions, EAT annotations that are useful for anchoring (that is, every but the *OTHER* class) were available for 38 questions. For 7 question no useful annotation was provided by AP or EAT (the EAT taxonomy used is described in a following section).

Table 1 shows that AP is not only more precise than EAT, it also achieves a similar if not slightly higher recall. Table 2 demonstrates that in questions where the AP was annotated, its performance is much better than that of the EAT.

This is somewhat expected because questions with an explicit focus seem intuitively easier to answer. These results allow us to distinguish between AP and EAT, and provide a viable reason for choosing to perform the branching with the prevalence of AP for focus extraction.

[6] Our corpus is more precisely described in a following section.

3 System Architecture

We approach both AP and EAT extraction with the same supervised machine learning technology based on the principle of maximum entropy [12][7].

The advantage of supervised machine learning approach over rule-based approach is to automatically construct a statistical model from the training data that can be used to classify unseen data according to a statistical model induced from the training data.

One disadvantage of supervised machine learning is that it requires training data that has to be annotated with the outcomes we expect to obtain while using the trained model on unseen data.

Machine learning also requires a careful selection of features that discriminate best between possible outcomes. Maxent[8] was used as the maximum entropy engine for EAT and AP extraction.

We annotated a corpus of 2100 questions. 1500 of those questions come from the Li & Roth corpus [7], 500 questions were taken from the TREC-10 questions and 100 questions were asked over the Italian Opera topic map.

The focus extractor operates as follows. First, the question is analyzed by means of tokenization. That is, words and punctuation marks are identified in the question string and put into a list of tokens. Spaces are discarded.

Next, the question is POS-tagged. That is, every word and punctuation mark is assigned a part-of-speech tag, e.g. noun, verb adjective.

Subsequently, a syntactic parse is created over the question that identifies relationships between its various constituents by associating articles with nouns that they determine or by assembling subjects, predicates and complements. The outcome of the syntactic parse is a tree structure that represents syntactic relationships in the question.

Finally, the AP extractor is fired and, if the AP is found, it is used as focus. Otherwise, the EAT extractor is used and its result is assigned to the focus.

While the tokenization is done by in-house software, POS-tagging and syntactic parsing are produced by OpenNLP.

3.1 AP Extraction

A model for extracting AP that is based on word tagging (i.e. AP is constructed on word level not on the question level) is proposed, as shown in Table 3.

Table 3. Sample annotation of APs

What	operas	did	Puccini	write	?
O	AP	O	O	O	O

[7] OpenNLP http://opennlp.sf.net, a popular software package for natural language research.

[8] http://maxent.sf.net

Our annotation guidelines limit the AP to the noun phrase that is expected to be the type of the answer. As such, it is different from the notion of focus as a noun *likely to be present in the answer* [3] or as *what the question is all about* [9]. For instance, a question such as *Where is the Taj Mahal?* does not yield any AP. Although the main topic is the Taj Mahal, the answer is not expected to be in a parent-child relationship with the subject. Instead, the sought after type is the EAT class LOCATION. This distinction is important for QA over semi-structured data where the data itself is likely to be hierarchically organized.

No AP is annotated for questions for which the answer is expected to be the topic mentioned in the question. For instance, *Who was Puccini?* has no AP. Indeed, the correct answer in the topic map is the topic *Puccini*, not a subtype of Puccini. Also, most questions requesting a synonym or an acronym explanation (*What does CPR stand for?*) are not marked up with an AP. Topic Maps design usually encourages authors to provide this kind of information (synonyms, abbreviations, lexical variations) directly in topic names.

On the other hand, descriptors are typical APs. A question such as *Who was J.F.K's wife?* would yield *wife* as AP. Annotation guidelines also specified that in descriptive nominal phrases (NPs), the most specific noun should be chosen so that the question *What was J.F.K's wife's name?* also yields to *wife* as AP, rather than name. APs typically occur as WH-complements (*What* ***country*** *was Puccini born in*) or as subjects denoting a supertype (*What are the* ***characters*** *appearing in Joan of Arc by Verdi?*), or eventually as objects (*Name a* ***philosopher***). Multiple APs could be annotated in a single question (*What* ***operas*** *and what* ***plays*** *did Puccini compose?*). Coordination (*What* ***operas*** *or* ***plays***) and modifiers – to the exception of relative clauses (*Name* ***Italian librettists*** *that lived between 1700 and 1800*) – were included as APs, but determiners and articles were excluded (*What were Christopher Columbus' three* ***ships****?*).

A final requirement was that APs had to be explicit. Adverbial WH-complements (e.g. How far) and key verbal phrases were as such excluded from APs. We devised a particular annotation scheme, the *asking point hint*, for key terms of the question that hint at a type but would require some form of lexical inference. For instance, *far* denotes a distance but would require further lexical inference to match a topic map entity. Similarly, *What is the oesophagus used for?* yields no AP, but *used for* could be extracted for type checking inference. Results for the *asking point hints* are not reported in this paper as we found that inter-annotator agreement was more difficult to achieve.

Asking points were annotated in 1095 (52%) questions out of 2100. These questions contained overall 19839 words, with 1842 (9.3%) of words marked as belonging to the asking point. The distribution of AP classes in the annotated data is shown in the Table 4.

A study of the inter-annotator agreement between two human annotators has been performed on a set of 100 questions. The Cohen's kappa coefficient [2] was at 0.781, which is lower than the same measure for the inter-annotator agreement

Table 4. Distribution of AP classes

Class	Count	%
AskingPoint	1842	9.3%
Other	17997	90.7%

on EAT. This is an expected result, as the AP annotation is naturally perceived as a more complex task. Nevertheless, this allows to qualify the inter-annotator agreement as good.

For each word, a number of features were used by the classifier, including strings and POS-tags on a 4-word contextual window. The WH-word and its complement were also used as features, as well as the parsed subject of the question and the first nominal phrase.

A simple rule-based AP extraction has also been implemented, for comparison. It operates by retrieving the WH-complement from the syntactic parse of the question and stripping the initial articles and numerals, to match the annotation guidelines for AP.

3.2 EAT Extraction

EAT is supported by a taxonomy of 6 coarse classes: HUMAN, NUMERIC, TIME, LOCATION, DEFINITION and OTHER. This selection is fairly close to the MUC typology of Named Entities which has been the basis of numerous feature-driven classifiers because of salient formal indices that help identify the correct class.

The class OTHER has been used for questions with less evident types and/or for some questions that are expected to be successfully analyzed by the AP extraction module and do not fall under any of the other categories. For instance, *Name the vessel used by the Atari Force in the DC comics* or *How do clouds form?* are both annotated as OTHER. In general, questions about manner, cause, comparison, condition, and other circumstantials are assigned the OTHER class.

The LOCATION class is assumed to only relate to geographic locations, thus excluding questions about e.g. body parts. DEFINITION questions are limited to the questions such as *What is Tosca*. The TIME class includes both date and time related question, or questions asking for a period or an interval.

We purposely limited the number of EAT classes to 6 as AP extraction already provides a fine-grained, dynamic classification from the question to drive the subsequent search in the topic map.

The distribution of EAT classes in the annotated data is shown in the Table 5.

A study of the inter-annotator agreement between two human annotators has been performed on a set of 200 questions. The resulting Cohen's kappa coefficient [2] of 0.8858 allows to qualify the inter-annotator agreement as very good.

We followed Li & Roth [7] to implement the features for the EAT classifier. Features included the first six words from each question taken literally, as well

Table 5. Distribution of EAT classes

Class	Count	%
TIME	136	6.5%
NUMERIC	215	10.2%
DEFINITION	281	13.4%
LOCATION	329	15.7%
HUMAN	420	20.0%
OTHER	719	34.2%

as their POS tags. The WH-complement at the start of the sentence had its own feature, except for the pairs *who-whose*, and *what-which* that were abstracted into two aggregate features.

Adjectives and adverbs occurring after *how* were also abstracted as a feature. The adverbs *many* and *much* occurring in the 2nd position in the question received each a feature, due to their frequency.

The forms of the verb *be* and *do* counted each for a feature, as well as the presence of verbs, nouns or adjectives at the 2nd position. The presence of an article at the 3rd position was used as a feature.

Four lists of words related to locations, people, quantities and time have been established, each wordlist being used as a feature to account for semantic similarity.

4 Evaluation

4.1 Evaluation Methods

The performance of the classifiers was evaluated on our corpus of 2100 questions annotated for AP and EAT. The corpus was split into 80% of training and 20% data. In order to account for variance, we ran the evaluation 10 times, shuffling the training and test data every run and calculating the sample standard deviation and the standard error of the mean.

Table 6 lists the figures for the accuracy of the classifiers, that is, the ratio between the correct instances and the overall number of instances. As the AP classifier operates on words while the EAT classifier operates on questions, we had to estimate the accuracy of the AP classifier per question, to allow for comparison. Two simple metrics are possible. A *lenient* metric assumes that the AP extractor performed correctly in the question if there is an overlap between the system output and the annotation on the question level. An *exact* metric assumes that the AP extractor performed correctly if there is an exact match between the system output and the annotation.

For example (Table 7), a question such as *What are Italian operas* leads to a lenient accuracy of 1, and an exact accuracy of 0. Precision for the AskingPoint class will be 1 and its recall will be 0.5.

Note that for overall focus evaluation, the exact metric was used.

Table 6. Accuracy of the classifiers

Accuracy	Value	Std. deviation	Std error
EAT	0.824	0.020	0.006
Lenient AP	0.963	0.020	0.004
Exact AP	0.888	0.052	0.009
Focus (AP+EAT)	0.827	0.020	0.006

Table 7. Evaluation example

	What	are	Italian	operas	?
Gold	O	O	AP	AP	O
Sys.	O	O	O	AP	O

Table 8. EAT performance by class

Class	Precision	Recall	F-Score
DEFINITION	0.887	0.800	0.841
LOCATION	0.834	0.812	0.821
HUMAN	0.902	0.753	0.820
TIME	0.880	0.802	0.838
NUMERIC	0.943	0.782	0.854
OTHER	0.746	0.893	0.812

4.2 Evaluation Results

Table 8 shows EAT results by class (with an overall accuracy of 0.824).

Tables 9 and 10 show AP results by class for the machine learning and for the rule-based classifiers.

Note that for AP, the *AskingPoint* F-score is more relevant than the overall extractor accuracy as the prevalence of *Other* tokens in the data set largely accounts for the high accuracy. The rule-based AP extractor performed at 0.536 F-score against 0.789 for the machine learning approach.

As shown in Figure 3, when AP was found, it was used for focus. During the evaluation, AP was found in 49.4% of questions. We call this number the *branching factor*.

It is expected that the focus accuracy, that is, the accuracy of the focus extraction system, is dependent on the performance of the AP and the EAT classifiers. We provide the formal reasoning below using the following symbols:

- TP_{AP} is the number of questions where AP was provided by the system and this was correct (true positive);
- FP_{AP} is the number of questions where AP was provided by the system and this was incorrect (false positive);
- TN_{AP} is the number of questions where AP was not provided by the system and this was correct (true negative);
- FN_{AP} is the number of questions where AP was not provided by the system and this was incorrect (false negative).

Table 9. AP Performance by class

Class	Precision	Recall	F-Score
AskingPoint	0.854	0.734	0.789
Other	0.973	0.987	0.980

Table 10. Rule-based AP performance by class

Class	Precision	Recall	F-Score
AskingPoint	0.608	0.479	0.536
Other	0.948	0.968	0.958

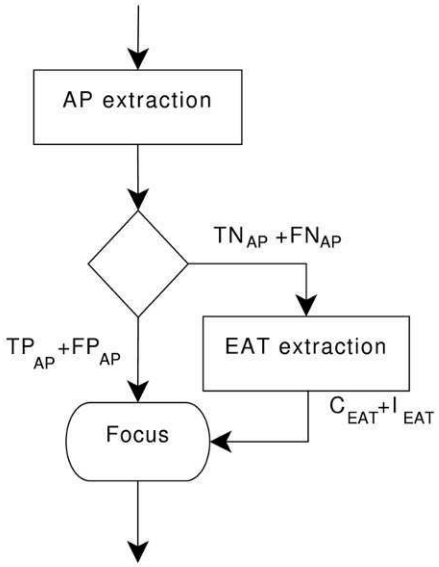

Fig. 3. Focus extraction flow diagram

Given N being the total number of questions, the branching factor, that is, the percentage of questions for which AP is provided by the system, is defined follows:

$$Y = \frac{TP_{\mathrm{AP}} + FP_{\mathrm{AP}}}{N}$$

The precision and recall of the *AskingPoint* class of the AP classifier is defined as

$$P_{\mathrm{AP}} = \frac{TP_{\mathrm{AP}}}{(TP_{\mathrm{AP}} + FP_{\mathrm{AP}})}$$

$$R_{\mathrm{AP}} = \frac{TP_{\mathrm{AP}}}{(TP_{\mathrm{AP}} + FP_{\mathrm{AP}})}$$

For the EAT classifier, we define

- $\mathrm{C_{EAT}}$ as the number of incoming questions that are assigned a correct class;
- $\mathrm{I_{EAT}}$ as the number of incoming questions that are classified incorrectly;

EAT branching accuracy is therefore

$$A_{\mathrm{EAT}} = \frac{C_{\mathrm{EAT}}}{C_{\mathrm{EAT}} + I_{\mathrm{EAT}}} = \frac{C_{\mathrm{EAT}}}{TN_{\mathrm{AP}} + FN_{\mathrm{AP}}}$$

It can be seen from Figure 3 that the sum $TP_{\mathrm{AP}} + C_{\mathrm{EAT}}$ represents the overall number of questions that were classified correctly. Therefore the overall accuracy, so the accuracy of the focus is

$$A_{\mathrm{FOCUS}} = \frac{TP_{\mathrm{AP}} + C_{\mathrm{EAT}}}{N}$$

This accuracy can be further developed to present the dependencies as follows:

$$A_{\mathrm{FOCUS}} = \frac{TP_{\mathrm{AP}} + C_{\mathrm{EAT}}}{N} = \frac{TP_{\mathrm{AP}}}{N} + \frac{C_{\mathrm{EAT}}}{N}$$

$$= \frac{TP_{\mathrm{AP}}}{(TP_{\mathrm{AP}} + FP_{\mathrm{AP}}) \,/\, \left(\frac{TP_{\mathrm{AP}} + FP_{\mathrm{AP}}}{N}\right)} + \frac{C_{\mathrm{EAT}}}{TN_{\mathrm{AP}} + FN_{\mathrm{AP}}} \left(\frac{TN_{\mathrm{AP}} + FN_{\mathrm{AP}}}{N}\right)$$

$$= \frac{TP_{\mathrm{AP}}}{(TP_{\mathrm{AP}} + FP_{\mathrm{AP}}) \,/ Y} + \frac{C_{\mathrm{EAT}}}{TN_{\mathrm{AP}} + FN_{\mathrm{AP}}} \left(\frac{N - TP_{\mathrm{AP}} - FP_{\mathrm{AP}}}{N}\right)$$

$$= \frac{TP_{\mathrm{AP}}}{TP_{\mathrm{AP}} + FP_{\mathrm{AP}}} Y + \frac{C_{\mathrm{EAT}}}{TN_{\mathrm{AP}} + FN_{\mathrm{AP}}} \left(1 - \frac{TP_{\mathrm{AP}} + FP_{\mathrm{AP}}}{N}\right)$$

$$= P_{\mathrm{AP}} Y + A_{\mathrm{EAT}}(1 - Y)$$

That is, the overall accuracy is dependent on the precision of the *AskingPoint* class of the AP classifier, the accuracy of EAT and the branching factor.

The formula above has been checked using the numbers from the 10-run evaluation. Given $\mathrm{TP_{AP}} = 1724$, $\mathrm{FP_{AP}} = 350$, $\mathrm{TN_{AP}} = 1841$, $\mathrm{FN_{AP}} = 285$, $\mathrm{C_{EAT}} = 1749$ and $\mathrm{I_{EAT}} = 377$, the AFOCUS was found to be identical to the reported value of 0.827.

The branching factor itself can be predicted using the performance of the AP classifier and the ratio between the number of questions annotated with AP and the total number of questions. We first define this ratio as

$$D_{\mathrm{AP}} = \frac{TP_{\mathrm{AP}} + FN_{\mathrm{AP}}}{N}$$

Using the ratio $\mathrm{D_{AP}}$, the branching factor is calculated as follows

$$Y = \frac{TP_{\mathrm{AP}} + FP_{\mathrm{AP}}}{N} = \frac{(TP_{\mathrm{AP}} + FN_{\mathrm{AP}})TP_{\mathrm{AP}}(TP_{\mathrm{AP}} + FP_{\mathrm{AP}})}{N(TP_{\mathrm{AP}} + FN_{\mathrm{AP}})TP_{\mathrm{AP}}} = \frac{D_{\mathrm{AP}} R_{\mathrm{AP}}}{P_{\mathrm{AP}}}$$

That is, if precision and recall are perfect the branching factor is equal to $\mathrm{D_{AP}}$.

5 Previous Work

Lehnert [5, 6] introduced the notion of focus (or asking point) for QA and described 13 conceptual question categories in a taxonomy later extended and grounded both theoretically and empirically by Arthur Graesser [4].

A few decades later, the term is still widely used but with somewhat various definitions. [3] characterised the question focus as "*a noun or a noun phrase likely to be present in the answer*" or at least nearby the answer (free text QA). [9] presented the focus as a mean to resolve the expected answer type when the WH-word is not specific enough. The focus is defined as "*a word or a sequence of words which define the question and disambiguate it in the sense that it indicates what the question is looking for, or what the question is all about*". For instance, *Which city has the oldest relationship as sister-city with Los Angeles* yields *city* as a focus. The focus of *Where is the Taj Mahal* is the *Taj Mahal*.

Our definition of *expected answer type* is similar to taxonomic question types found in previous work. On the other hand, our notion of *asking point*, is slightly different from what has previously been called a focus/asking point. In particular, the AP extraction is specifically aimed at the retrieval of a parent type in an ontology, rather the identification of the central topic of the question.

In open domain QA, machine learning approaches to question classification have proved successful since Li & Roth [8], who reported state of the art results with a hierarchical classifier guided by layered semantic types of answers (92.5% accuracy on coarse classes tested over 1000 TREC questions and 89.3 % for 50 fine-grained classes, trained on 21,500 questions). Features include syntactic information about the question as well as semantic features (Named Entities, list of words grouped by similarity and by semantic classes).

[1, 10] describe MOSES, an multilingual QA system delivering answers from Topic Maps. MOSES extracts a focus constraint defined after [13] as part of the question analysis, which is evaluated to an accuracy of 76% for the 85 Danish questions and 70% for the 83 Italian questions. The focus is an ontological type dependent on the topic map, and its extraction is based on hand-crafted rules. In our case, focus extraction – though defined with topic map retrieval in mind – stays clear of ontological dependencies so that the same question analysis module can be applied to any topic map.

6 Future Work and Conclusion

We presented a question classification system based on our definition of *focus* geared towards QA over semi-structured data where there is a parent-child relationship between answers and their types. The specificity of the focus degrades gracefully in the approach described above. That is, we attempt the extraction of the AP when possible and fall back on the EAT extraction otherwise.

We identify the focus dynamically, instead of relying on a static taxonomy of question types, and we do so using machine learning techniques throughout the application stack.

A mathematical model has been devised to predict the performance of the focus extractor.

Despite using similar features, the F-Score (0.824) for our EAT classes is slightly lower than reported by Li & Roth [8] for coarse classes. We may speculate that the difference is primarily due to our limited training set size (1,680 questions versus 21,000 questions for Li & Roth). On the other hand, we are not aware of any work attempting to extract AP using machine learning in order to provide dynamic classes to a question classification module. We are currently analysing the errors produced by EAT and AP extractors to improve on performance.

On-going work also includes working on the exploitation of the results provided by the focus extractor in the subsequent modules of the QA over Topic Maps, namely anchoring, navigation in the topic map, graph algorithms and reasoning.

7 Acknowledgements

This work has been partly funded by the Flemish government (through IWT) as part of the ITEA2 project LINDO (ITEA2-06011).

References

1. P. Atzeni, R. Basili, D. H. Hansen, P. Missier, P. Paggio, M. T. Pazienza, and F. M. Zanzotto. 2004. Ontology-Based Question Answering in a Federation of University Sites: The MOSES Case Study. In NLDB, pages 413–420.
2. J. Cohen. 1960. A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20, No.1:37–46.
3. O. Ferret, B. Grau, M. Hurault-Plantet, G. Illouz, L. Monceaux, placeI. Robba, and A. Vilnat. 2001. Finding an Answer Based on the Recognition of the Question Focus. In 10th Text Retrieval Conference.
4. A. C. Graesser, C. L. McMahen, and B. K. Johnson. 1994. Question Asking and Answering. In M. A. Gernsbacher, editor, Handbook of Psycholinguistics, pages 517–538. San Diego: Academic Press.
5. W. Lehnert. 1978. The Process of Question Answering. Lawrence Erlbaum Publishers, Hillsdale, N.J.
6. W. Lehnert. 1986. A Conceptual Theory of Question Answering. Readings in natural language processing, pages 651–657.
7. X. Li and D. Roth. 2002. Learning Question Classifiers. In 19th International Conference on Computational Linguistics (COLING), pages 556–562.
8. X. Li and D. Roth. 2006. Learning Question Classifiers: The Role of Semantic Information. Journal of Natural Language Engineering, 12(3):229–250.
9. D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Goodrum, R. Girju, and V. Rus. 1999. LASSO: A Tool for Surfing the Answer Net. In 8th Text Retrieval Conference.
10. P. Paggio, D. H. Hansen, R. Basili, M. T. Pazienza, and F. M. Zanzotto. 2004. Ontology-based question analysis in a multilingual environment: the MOSES case study. In OntoLex (LREC).

11. S. Pepper. 2000. The TAO of Topic Maps. In Proceedings of XML Europe 2000.
12. A. Ratnaparkhi. 1998. Maximum Entropy Models for Natural Language Ambiguity Resolution. Ph.D. thesis, University of Pennsylvania, Philadelphia, StatePA.
13. M. Rooth. 1992. A Theory of Focus Interpretation. Natural Language Semantics, 1(1):75–116.