

Answer Comparison: Analysis of Relationships between Answers to ‘Where’-Questions

Tiphaine Dalmas and Bonnie Webber

Institute for Communicating and Collaborative Systems (ICCS),
School of Informatics, University of Edinburgh,
2 Buccleuch Place, Edinburgh EH8 9LW, Scotland, UK.

t.dalmas@sms.ed.ac.uk bonnie@inf.ed.ac.uk

Abstract

For many reasons, Question Answering must deal with questions that have multiple answers. To this end, we have built a framework for answer comparison based on a technique using information fusion that has successfully been applied in automated summarization. The architecture of this system is a direct application of the Model-View-Controller design pattern which allows us to define an answer in terms of content, structure and rendering. We present an experiment using this system on TREC ‘where’-questions and analyse relationships discovered between potential answers. We show that this approach is robust and a first step towards complex/structured answer generation.

1 Introduction

Interest in automated Question Answering (QA), especially large-scale open-domain QA, has been stimulated by the annual Text REtrieval Conference¹ organized by NIST. The task has evolved since the first official QA track launched in 1999: There is now a sharp distinction between factoid and list/definition questions. Only one answer is allowed for the first kind, and it has to be an *exact* answer (Voorhees, 2002) whereas originally five strings from 50 to 250 bytes were allowed. For list/definition questions such as *What grapes are used in making wine?* or *Who is Nostradamus?*, systems must provide either an enumeration or multiple answers covering different sides of the topic. The QA roadmap proposed by Burger et al. (2002) goes even further by asking eventually for a generated summary instead of a flat list of extractions.

The question we asked ourselves was whether analysing the *set* of extractions (potential answers), prior to choosing or constructing the answer(s) to be output, would help with all three kinds of questions.

In this context, we built a framework, QAAM (QA Answer Model), based on the Model-View-Controller (MVC) design pattern which we think gives an engineering definition for an answer that fits exactly the QA task. The model generation technique we use has been inspired from recent developments in multi-document summarization (Mani and Bloedorn, 1998; Barzilay et al., 1999) and makes use of information fusion to create a model that relates potential answers to one another.

The difficulty is that QA is not a 100% confident technology yet. Many of the extractions (potential answers) are still incorrect. Given this, does it make sense to analyse the full set of extractions? Before working on the generation of complex and structured answers, we wanted to check first the feasibility of such a framework on the results given by current QA systems: It might be that incorrect answers are actually too noisy for the answer model to be robust. In this paper, we analyse the results of a first experiment on factoid questions (‘where’-questions). We tested QAAM for reranking to see if information fusion was not harmed by incorrect answers and would still help answer selection. This experiment also provided an important number of relationships between answers we analyse here.

2 QAAM

2.1 A MVC approach

We construe the concept of *answer* within an engineering approach to QA based on the MVC pattern of user interface design (Krasner and Pope, 1988). This approach allows us to distinguish the two main parts that define an answer: its content and structure (model) and its rendering (view). This allows a system to make separate choices about them: Whether to give a precise or an elaborate answer, versus whether to present the answer graphically or textually. For example, TREC currently requires an exact answer and the most expected one, and it re-

¹<http://www.trec.nist.gov>

quires that answer to be output as a string along with the source document as justification. Taking QA as a user interface problem, in general, allows us to address two points:

(1) *The amount and kind of information to be presented.* For example, when do we provide an intensional description rather than an extensional enumeration? For example, to *What animals carry their young in a pouch?*, an intentional answer would be *marsupials*, an extensional answer *kangaroos, wombats, opossums*. How can we allow a user to influence this rendering according to the kind of information they are looking for, e.g. a detailed answer, a short answer, the most frequent answer or all answers?

(2) *The modality in which to cast the presentation.* There are many, even simple, factoid questions for which text alone is not the best medium (Andre and Thomas, 1994). For instance, the answer to a question such as *Where are diamonds mined?* would probably be best rendered with a map, while the answer to *Who was Galileo?* might best be rendered with a combination of a biographic summary and pictures. Although the main problem of the QA community is still how to *obtain* correct answers rather than how to *render* them, it is still a worthwhile and interesting question to pursue. An answer could be provided as a generated text (Burger et al., 2002) or as a map (Leidner et al., 2003), or if the user is another computing system, one would prefer to provide the answer as a formal structure rather than text.

To support such different views for a single question, we have been making use of the MVC design pattern (Figure 1).

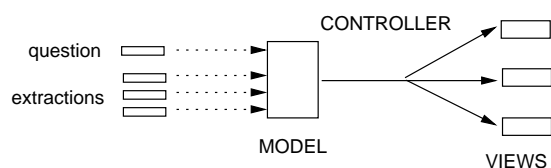


Figure 1: MVC approach to QA

Starting with a list of extractions and a question, we build a *model* representing an answering space.

Once a model is built and the answering space structured, different *views* can be derived to render it. This distinction between model and view fits exactly the QA task, expressing accurately that an answer could be phrased and presented in different

manners, but always according to the data the system has found.

The role of the *controller* in this framework is to retrieve the final answer according to the end user requirements in terms of answer properties or features. In actual user interface application, the role of the controller is more interactive, we believe that it could be the case in dialog-oriented QA. The controller allows users to influence a rendering according to the degree of information required, e.g. a detailed answer or a short answer. The controller can be seen as a function that defines the content selection and its rendering.

In this generic framework, we propose a techniques based on information fusion to generate a model. But whatever technique is chosen, we think that applying the MVC to QA is particularly relevant to this task.

2.2 A model based on information fusion

QAAM takes as input a question and a collection of the corresponding extractions, all of which are assumed to have been tokenized and tagged with Part-of-Speech (POS)². From this, the model (a directed graph) is built in two steps.

The first step generates a set of nodes from the question and the extractions. Nodes are projected on the basis of annotation of the source text. A node may correspond to a token, a word or a multi-word expression. A projection is considered to be an entity that is potentially *involved in* and *relevant to* a local knowledge representation. That is, a node (projection) may not be an answer in itself but rather, related to the answer³. Nodes are represented as features (i.e. attribute-value pairs of linguistic annotation) which can then be used during the comparison process. For instance, the extraction *Europe* generates the following node:

```

{ token = Europe
; pos = NNP
; src = a.0.0
}
  
```

POS stands for the Part-of-Speech tag, src keeps track of the source of the extraction, e.g. a for answer candidate, extraction #0, offset #0. This is to distinguish between question nodes and answer nodes and eventually to allow further strategies based on node adjacency. In our experiment, nodes

²We use our own tokenizer and MXPOST for POS tagging (Ratnaparkhi, 1996)

³We therefore prefer the term 'information fusion' over 'answer fusion' (Girju, 2001).

were projected from nominal phrases (NPs) from the question and the answer extractions, by matching token and POS n-grams (regular expressions). Most extractions in TREC are NPs and projecting them was actually sufficient to cover our dataset.

The second step uses different resources (WordNet, abbreviation/acronym recognition) to discover relationships between nodes, which are used to label edges between them. We follow on Webber et al. (2002) who characterize four broad categories of relations between answers: *equivalence*, *inclusion*, *aggregation* and *alternative*. We implemented the discovery of the two first only: Deciding whether the relationship between two nodes should be characterised as *Aggregation* or *alternative* requires a deeper analysis of the question. For instance, *Where is Perth?* expects only *alternative* answers (i.e. each answer may be the location of a different Perth) but the form of the question does not completely predict the expected relationship among such answers. The similar question *Where is the Euphrates River?* allows a conjunction of answers (*aggregation*) since rivers can cross several countries. Thus the system is currently restricted to *equivalence* and *inclusion*.

Figure 2 provides a complete view⁴ of the answer model generated from the following set of question and extractions:

What continent is Scotland in? (TREC, 1647)

- | | |
|--------------|--------------|
| 1. Europe | 4. Ireland |
| 2. EDINBURGH | 5. in Africa |
| 3. Africa | 6. Scotland |

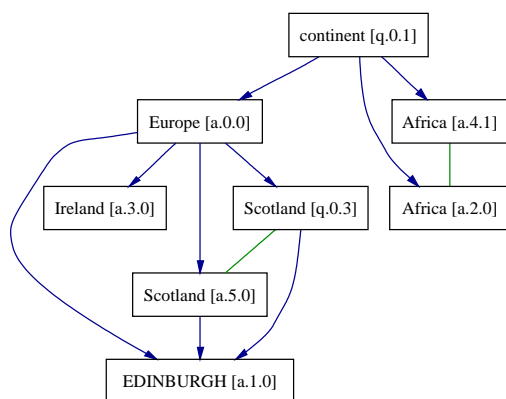


Figure 2: Complete view of the answer model generated for the question answering set *What continent is Scotland in?* (before the transitive closure).

⁴We used the graph visualization tool Graphviz (Gansner and North, 1999)

(In all figures, edges without arrows stand for *equivalence* links, the others for *inclusion*).

The last step consists in performing the transitive closure of our relations over the graph.

From this model, different views can be automatically derived by querying the graph. For instance, a ‘TREC view’ asks for an extraction, thus the system will simply output the `token` feature of the node (rendering mode). To select an *exact* answer (content specification), a single node will be chosen that is as *specific* as possible. *Specificity* is a property that can be translated into a graph query using the inclusion relation (the fewer children a node has, the more specific it is likely to be, e.g. *Edinburgh* is the most specific node in the graph given above). Redundancy can be translated by the equivalence relation, in the same graph, *Scotland* and *Africa* are two redundant nodes. We refer to Dalmas and Webber (2003) for the detailed list of such properties that we combined to get the view required by this experiment.

3 Experiment and results

We performed this first experiment to check the feasibility of such a system over real sets of question/answers pairs. Because incorrect answers are still frequent, we wanted to check whether 1) it was feasible to discover relations among answers and 2) incorrect answers would not harm the robustness of the final answer selection.

We used 85 sets of question/answers pairs built from the judgement files provided by NIST, covering TREC 8 to 11. These sets contains at least 5 incorrect answers and 1 to up to 5 correct answers for each question, which makes a realistic distribution of incorrect answers.

Since we wanted to compare QAAM to a more traditional approach that does not make use of relations among answers, we chose those ‘where’-questions for which there is an obvious relationship between the question and the answer, a spatial inclusion such as *Where is Glasgow?*, *Glasgow is in Scotland*, that can be easily checked in WordNet. We also started on typical factoid questions to show that there are actually multiple answers to such questions (although TREC rules oppose them to list/definition questions) and that one can make use of them to improve answering.

We defined the task as a reranking problem. The system takes as input a question and a list of extractions, generates a graph and outputs a ‘TREC view’,

i.e. an ordered list of strings, each corresponding to different node. The highest ranked string is then evaluated against the TREC answer patterns.

We compared two different views:

- QAAM-1 represents the baseline based on a traditional approach which checks the relation between the question and each answer. This approach draws on the research done on the notion of *answer type*, i.e. does the candidate answer respects the requirements expression by the type of question being asked.
- QAAM-2 makes use of information fusion: all the relations are checked out including those between answers (QAAM-2 can be seen as an extension of QAAM-1).

(Dalmas and Webber (2003) details the two approaches and specifically the answer properties combined for each strategy).

For instance, in the following answer model, QAAM-1's only option is a random selection because there is no relation available between a question node and an answer node. Contrariwise QAAM-2 is helped by the inclusion discovered between *Egypt* and *Luxor*.

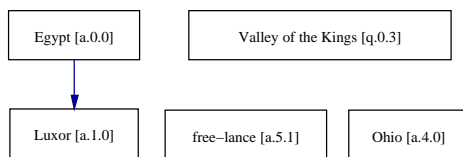


Figure 3: Answer model generated for *Where is the Valley of Kings?*

For the 85 questions, the top-ranked answer from QAAM-1 was correct in 42 cases (49%), with a Mean Reciprocal Rank of 0.63. QAAM-2 ranked correctly the first answer in 62 cases (72%), with a MRR of 0.82 (the MRR score evaluates the overall ranking method: $\sum \frac{1}{r}$, where r is the rank of the first correct answer for each question).

The analysis of errors made respectively by QAAM-1 and QAAM-2 is detailed in Dalmas and Webber (2003). The data set and the answer models are all available at <http://www.iccs.informatics.ed.ac.uk/~s0239548/work.html>. In the next section, we focus on the answer models generated by QAAM-2 and specifically on the distribution of relationships among nodes.

4 Analysis of relationships

The experiment generated 85 answer models with an overall count of 841 nodes (16.64% from questions and 83.35% from extractions) and 785 relations. Figure 4 shows the number of relations between question nodes only (q/q), question nodes and answer nodes (q/a) and among answer nodes (a/a).

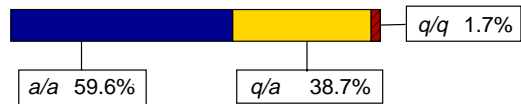


Figure 4: Relations repartitions

There can be a few relations among the nodes projected from the question (q/q). For instance, the model draws an inclusion between the two nodes *Rome* and *Italy* projected from *What river runs through Rome, Italy?* (TREC, 1836).

The proportion of relations among answers only (a/a) is the largest one, which may give a clue why QAAM-2 performed better than QAAM-1.

Table 1 indicates the repartitions of inclusion and equivalence relationships per question.

Table 1: Analysis of relations types per question

	equivalence			
	min	max	average	total
q/q	0	0	0	0
a/a	0	26	2.05	175
q/a	0	7	1.20	102
overall	0	30	3.25	277

	inclusion			
	min	max	average	total
q/q	0	1	0.15	13
a/a	0	37	3.45	293
q/a	0	12	2.37	202
overall	0	44	5.97	508

Min and *max* indicate the minimum and maximum number of times the given relation appears in the answer model of a question. For 6 of the 85 questions, their graphs were actually empty of relationships (which is why 0 is the minimum of each relation type). The row labelled *overall* ignores what the relation holds between (q or a) and just gives the minimum and maximum number of times that the relation appears in the answer graph

of a question. So 0 is still the minimum and the graph with the largest number of equivalence relations contained 30 of them.

Most of QAAM’s relation discovery consists in the inclusion relation, which relates to the kind of question we focused on (‘where’-questions inducing an entailment or an inclusion with their answers).

Inclusion is also the main relation found among answers. Previous studies on answer reranking were mainly based on the computation of answer frequencies. For example, Brill et al. (2001) and Clark et al. (2001) both used a simple string matching to compute redundancy among answers, i.e. a rough equivalence. Although our data set is biased because restricted to spatial inclusion, it is clear that it is worth exploiting other kinds of relations among answers.

The next table describes the same repartition but takes into account the type of the answer node, either correct (ca) or incorrect (ia).

Table 2: Analysis of relations involving correct versus incorrect answer nodes

	equivalence			
	min	max	average	total
ca/ca	0	20	0.91	78
q/ca	0	0	0	0
q/ia	0	7	1.20	102
ia/ia	0	16	0.98	83
ca/ia	0	8	0.16	14
overall \neg /ia	0	28	2.34	199

	inclusion			
	min	max	average	total
ca/ca	0	5	0.3	26
q/ca	0	10	1.49	127
q/ia	0	6	0.88	75
ia/ia	0	21	1.23	105
ca/ia	0	29	1.91	162
overall \neg /ia	0	40	4.02	342

As expected the only relation involving a question node and a correct answer node (q/ca) is the inclusion. The data were based on questions requesting an entailment or an inclusion, thus the equivalence relation always involves an incorrect node. In *What county is Modesto, California in?* (TREC, 895), *California* was linked to an equivalent node *Calif.* which is correct but would only be useful if

the question were *What state is Modesto, California in?* (similar to the old joke *Who is buried in Grant’s tomb?*)

The inclusion relation can also involve incorrect nodes, especially if the question contains a frequent NP such as *capital* in *What is the capital of Ethiopia?* (TREC, 1161). Such nodes are frequently linked to other nodes because they are classifiers in geographical knowledge. In this case, an incorrect node, *London*, was linked to the question node projected from *capital*. Although this looks inconvenient, it helped QAAM-2 by enlarging the partition of the graph relating to capitals, which actually contains the correct answer.

Figure 5 describes the overall distribution of correct and incorrect answer nodes among relations.

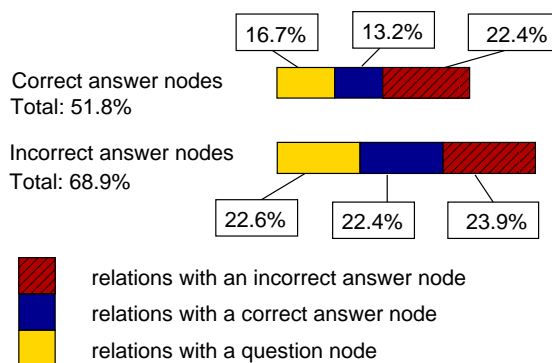


Figure 5: Relations repartitions of correct versus incorrect answer nodes

The count of relations involving incorrect answer nodes is actually massive ($q/ia + ia/ia + ca/ia = 541$, representing 68.9% of the total number of relations). The distribution of relations between correct answer nodes only (ca/ca) is small (13.2%). On average there were only a few inclusion relations between them (0.3, Table 2). It might be that incorrect but related answers actually helped. If we had an oracle indicating which answer nodes are correct or incorrect so that the system only draws relations between question nodes and correct answer nodes, only 29.42% ($ca/ca + q/ca = 231$) of the current relations would have been inferred.

To check the role of incorrect answer nodes, we carried out another experiment with such an oracle. Figure 6 below is the graph generated by QAAM with the oracle for the question-answer set given in Figure 2. This can be compared to the answer graph in Figure 2 that has been generated without oracle.

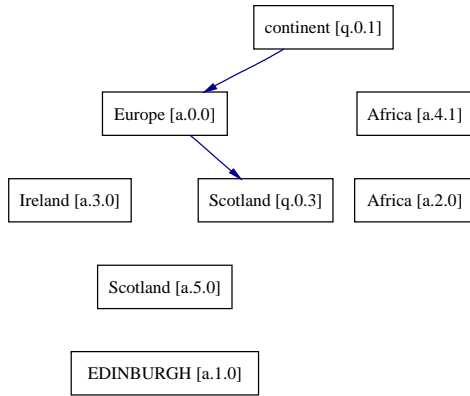


Figure 6: *What continent is Scotland in?* QAAM graph helped by an oracle.

The intuition behind this experiment was that if it could improve QAAM-1 by blocking relations between question nodes and incorrect answer nodes, it would also lower QAAM-2 results by influencing features such as the partition size of a node (the number of related neighbours plus the node itself).

Results shown in Table 3 show that the use of an oracle improved significantly QAAM-1. However, results for QAAM-2 were unexpected. It too showed improvement, though not as large an improvement as QAAM-1. This shows that a strategy that considers relations among both questions and answers is not only better but more robust and resistant to incorrect answers than a strategy that considers only relations between questions and answers alone.

Table 3: Comparative results for reranking

	QAAM-1	
	first-rank score	MRR
standalone	49%	0.63
with an oracle	65%	0.71

	QAAM-2	
	first-rank score	MRR
standalone	72%	0.82
with an oracle	78%	0.85

5 Conclusion

This experiment was performed over 85 questions only and of a specific type, thus it is difficult to generalize these results. But within this experiment we set up a first framework for performing answer comparison and evaluate our strategy.

Our assumption was that extractions given by current QA systems are not random strings: They have been selected through an elaborate filtering process including information retrieval and query reformulation, shallow or deep parsing, answer pattern matching and eventually more advanced techniques based on first-order logic. Extractions are thus somehow *related*. Our experiments showed:

1. Extractions are indeed related and it is feasible to discover these relations at low cost, i.e. using shallow methods (the highest level of parsing is POS tagging).
2. Making use of multiple answers and their relations is helpful and more robust.
3. Our modeling has the advantage of providing intuitive criteria based on graph properties for defining the characteristics of the sought-after answer (specificity for instance).

Finally, although our experiments did not test this, it should be clear from the graphs that, if answers can be related, it is possible to provide more structured answers and go beyond answers as extractions or flat lists of extractions.

As our next targets, we would like to extend this experiment to other question types and on larger data sets. Currently, QAAM-2 ranks potential answers heuristically, based on a combination of nodes and sub-graph properties, in order to identify the best ‘TREC view’ answer. Heuristics include disfavoring nodes with a greater number of included children. The size of the partition is also a good indicator. In Figure 7 below, among the four partitions discovered (A, B, C, D), the largest one contains the correct answers, *Scotland* and *Britain*. We also computed for each node how many question nodes were equivalent or included, directly or by using transitivity along the graph.

In future work, we want to use these properties as a feature space for supervised machine learning. Each instance would be a node, correct or not, with a set of attributes, either numeric (number of children, number of equivalent nodes, etc ...) or symbolic (POS tag, token). With a larger data set, we could apply machine learning techniques on graph properties to generalize this experiment to other question types and improve node selection. By introducing the question type as a feature, it could be possible to learn what graph topology is generated for a certain type of question, e.g. inclusion plays an

important role for ‘where’-questions, equivalence may be more represented in answer models generated to ‘synonym’ or abbreviation questions, such as *What does NASA stand for?* or *What is another name for the North Star?*

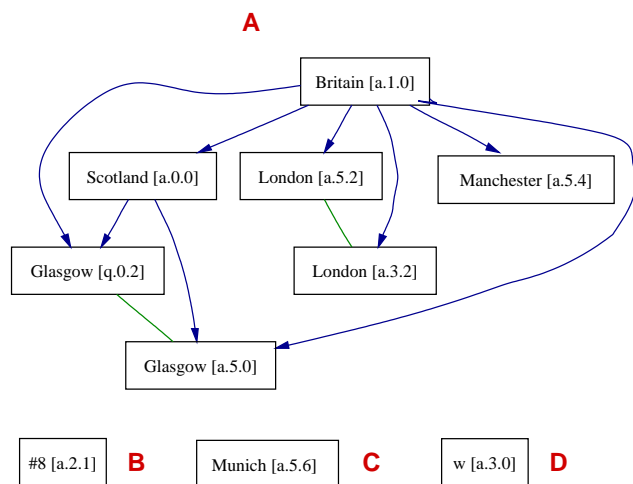


Figure 7: *Where is Glasgow?*. The answer model is partitioned in 4 subgraphs: A, B, C and D. A contains the correct answers, *Scotland* and *Britain*

We also plan to evaluate answers based on a partition of the graph, i.e. a set of nodes and their relations. In Figure 7, a structured answer could be generated as *Glasgow is in Scotland, Britain*. This would be a first step towards complex/structured answer evaluation.

Another important target is to extend the set of tools that can be exploited in relation discovery. In this experiment, WordNet covered most of the knowledge required to answer geographical questions. We would like to develop a collaborative discovery system based on a set of tools each highly specialized in one relation. This would draw on recent research on paraphrase recognition (equivalence) and inclusion discovery (for instance meronymy, (Girju et al., 2003) or automatic extraction of geographical knowledge (Ourioupina, 2002)).

References

E. Andre and R. Thomas. 1994. Referring to world objects with text and pictures. In *COLING-94*, pages 530–534.

R. Barzilay, K. R. McKeown, and M. Elhadad. 1999. Information fusion in the context of multi-

documents summarization. In *Proc. of the 37th Association for Computational Linguistics*, pages 550–557.

E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng. 2001. Data intensive question answering. *Proceedings of 10th TREC*.

J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C. Lin, S. Maiorano, G. Miller, D. Moldovan, B. Ogdan, J. Prager, E. Riloff, A. Singhal, R. Shrihari, T. Strzalkowski, E. Voorhees, and R. Weischedel. 2002. Issues, tasks and program structures to roadmap research in question and answering. *NIST*.

C.L.A. Clark, G.V. Cormack, and T.R. Lynam. 2001. Exploiting redundancy in question answering. *SIGIR 2001*.

T. Dalmas and B. Webber. 2003. Information fusion for answering factoid questions. *2nd CoLogNET-ElsNET Symposium. Questions and Answers: Theoretical and Applied Perspectives*.

E. R. Gansner and S. C. North. 1999. An open graph visualization system and its applications to software engineering. *Software – Practice and Experience, 00 (S1), 1-5*.

R. Girju, A. Badulescu, and D. Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. *Proceedings of Human Language Technology*.

R. Girju. 2001. Answer fusion with on-line ontology development. *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL) - Student Research Workshop*.

G. Krasner and S. Pope. 1988. A cookbook for using the model-view-controller user interface paradigm in smalltalk -80. *Journal of Object-Oriented Programming (JOOP)*.

J. Leidner, G. Sinclair, and B. Webber. 2003. Grounding spatial named entities for information extraction and question answering. *Proceedings of HLT-NAACL 2003 Workshop on Geographical References*.

I. Mani and E. Bloedorn. 1998. Summarizing similarities and differences among related documents. *Information Retrieval, 1, 35-67, 1999*.

O. Ourioupina. 2002. Extracting geographical knowledge from the internet. *International Workshop on Active Mining*.

A. Ratnaparkhi. 1996. A maximum entropy part-

- of-speech tagger. *Proc, Empirical Methods in Natural Language Processing Conference*.
- E.M. Voorhees. 2002. Overview of the TREC 2002 question answering track. *Proceedings of 11th TREC*.
- B. Webber, C. Gardent, and J. Bos. 2002. Position statement: Inference in question answering. In *Proceedings of the LREC Workshop on Question Answering: Strategy and Resources*, pages 19–26, Las Palmas.