D4.2.2: Final Response Component

Tiphaine Dalmas

Distribution: Public

SpaceBook

Spatial & Personal Adaptive Communication Environment: Behaviors & Objects & Operations & Knowledge

270019 Deliverable 4.2.2

August 31, 2013

Project funded by the European Community under the Seventh Framework Programme for Research and Technological Development

The deliverable identification sheet is to be found on the reverse of this page.

Project ref. no.	270019
Project acronym	SpaceBook
Project full title	Spatial & Personal Adaptive Communication Environment: Behaviors & Objects & Operations & Knowledge
Instrument	STREP
Thematic Priority	Cognitive Systems, Interaction, and Robotics
Start date / duration	01 March 2011 / 36 Months

Security	Public
Contractual date of delivery	M30 = August 2013
Actual date of delivery	August 31, 2013
Deliverable number	4.2.2
Deliverable title	D4.2.2: Final Response Component
Type	Prototype
Status & version	Final 1.0
Number of pages	12 (excluding front matter)
Contributing WP	4
WP/Task responsible	UE
Other contributors	
Author(s)	Tiphaine Dalmas
EC Project Officer	Franco Mastroddi
Keywords	

The partners in SpaceBook are:

Umeå University	UMU
University of Edinburgh HCRC	UE
Heriot-Watt University	HWU
Kungliga Tekniska Högskola	KTH
Liquid Media AB	LM
University of Cambridge	UCAM
Universitat Pompeu Fabra	UPF

For copies of reports, updates on project activities and other SPACEBOOK-related information, contact:

The SPACEBOOK Project Co-ordinator:
Dr. Michael Minock
Department of Computer Science
Umeå University
Sweden 90187
mjm@cs.umu.se
Phone +46 70 597 2585 - Fax +46 90 786 6126

Copies of reports and other material can also be accessed via the project's administration homepage, `http://www.spacebook-project.eu`

Contents

Executive summary

This report documents the Question Answering (QA) component of the SPACEBOOK project corresponding to Task 4.2.2 delivered in month 30. We describe the current features supported by QA and provide further analyses of the evaluation ran in the streets of Edinburgh in 2012. We review changes contributed since then.

1 Introduction

This report documents the recent changes contributed to the Question Answering (QA) component developed to handle exploration tasks in SPACEBOOK. QA is defined in Task T4.2.2:

Explore and evaluate methods for generating, validating and embedding answer snippets from geo-tagged documents within the city data model. Methods to be considered include both those used in situated text generation and those used in identifying textual material to answer definition questions in QA in particular, to recognise what information is both significant and novel, given the pedestrians interests and trajectory.

Although a "response" component, QA could be more appropriately seen as a "content" provider: it connects the City Model (CM) to unstructured data sources to automatically provide additional information about points of interests, and also delivers knowledge on various topics relating to Edinburgh but out of the City Model's scope. It is used by the Interaction Manager (IM) to directly answer the user's questions, or to push information relevant to his trajectory and/or interests.

QA has been integrated to SPACEBOOK and evaluated in Edinburgh's street experiments. Deliverable D6.2.2 provided a detailed analysis of QA performance (overall accuracy, component-based performance and qualitative feedback from subjects). In this document, we first provide an overview of the QA work flow between components as it currently stands, and then review changes that have been contributed since the evaluation.

2 QA architecture and work flow

QA consists of 4 major modules: (1) question classification, (2) focus (asking point) extraction, (3) co-reference checking and resolution, and (4) search. Figure 1 presents the flow between components.

Because QA delivers content from unstructured data (as opposed to the CM database), it has its own request analysis component based on open domain techniques for textual search (Li & Roth, 2002; Mikhailian *et al.*, 2009). When a question comes in, it is first classified in one of four categories:

1. `out-of-scope` - navigation queries or question type not covered by QA:
   Where is the Royal Mile? Right or left? How much is a ticket?,
2. `biography` - "definition" questions about people:
   Who was John Knox? What is he famous for?
3. `description` - other definition questions:
   What is haggis? Tell me about John Knox House. What is this statue?,
4. and `next-segment` questions seeking further information on a previous topic:
   Tell me more..

`Biography` and `description` questions are passed through the focus extractor in charge of pin-pointing the asking point: *Who was **John Knox**? What is **he** famous for? What is this **statue**?*.

Once the focus is extracted, QA checks whether it is a co-referring expression or not.

If the focus is a co-referring expression, QA will attempt to resolve it to either an anaphoric candidate (from dialogue history) or a deictic entity (using the City Model visibility engine).

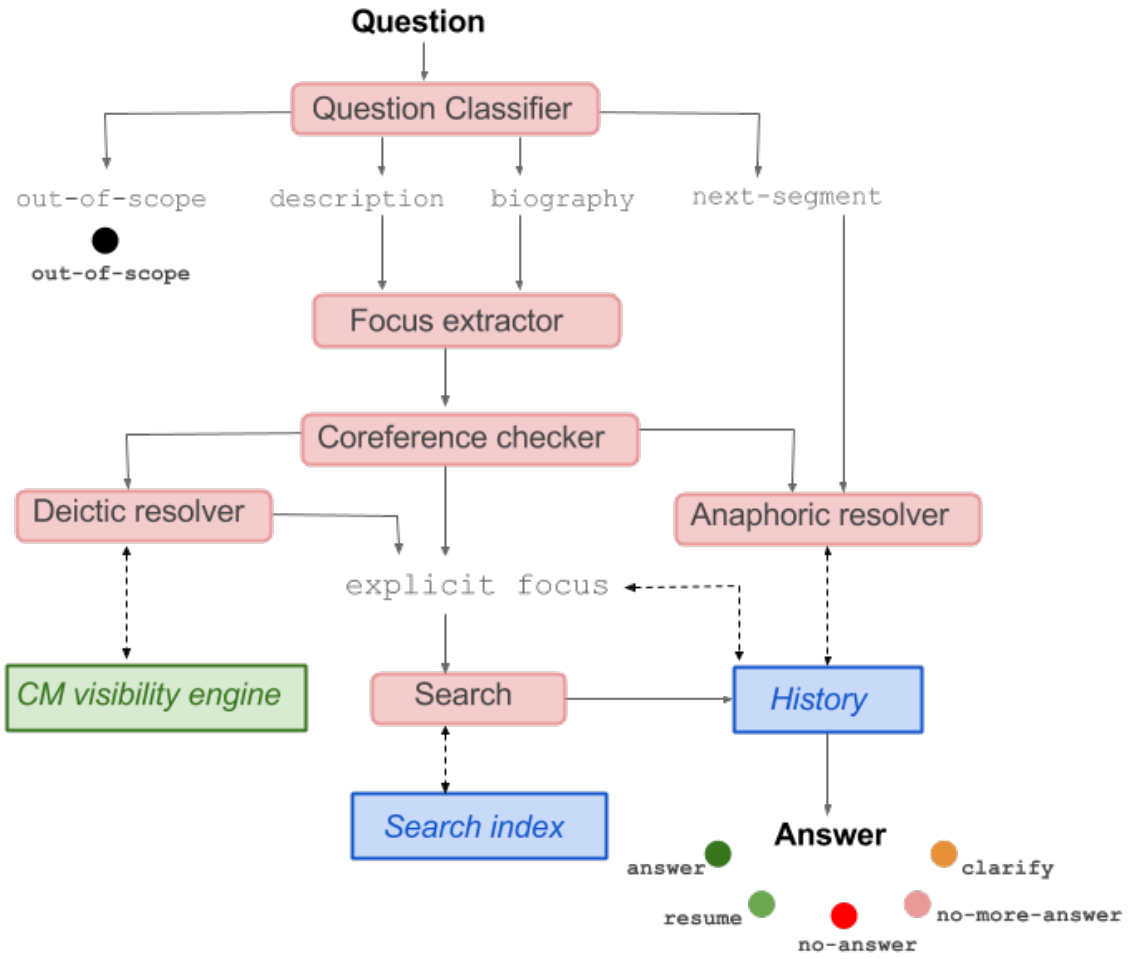

Figure 1: QA flow across components

If the focus is explicit or has been made explicit via co-reference resolution, QA checks its history to see if the entity has been talked about previously, in which case it will resume information from history. If not, a search will be performed.

The answer is added to the dialogue history before being sent back.

QA possible outcomes are encoded in 6 dialogue acts:

1. `out of scope`: QA does not recognise the input as an utterance it can process.
2. `no answer`: no information has been found for the question's focus.
3. `answer`: an answer has been found for a new focus, the first segment is provided.
4. `resume`: the question's focus has been previously talked about, QA resumes on the next segment.
5. `no more answer`: the question's focus has been previously talked about, but QA does not have further information.

6. `clarify`: QA detected a co-referring question but could not resolve it.

The answer is passed via the IM to the NLG component in charge of producing the system response.
In the next sections, we review the changes that have been contributed to QA since the last evaluation: we will start with updates concerning data sources, and pursue with features that have been added to the QA API, in particular topic tagging and deictic resolution.

3 Data sources updates

This section reviews the contribution of the data sources during the street evaluation, and two updates performed in that area: (1) removing WordNet as a data source, and (2) text simplification for TTS.
The following sources were used to provide content during the street evaluation:

- Entries from The Gazetteer for Scotland (Gittings, 2012): 325 points of interest and 391 people of interest.
- Introductory texts from Wikipedia entries limited to the city's network. This network was computed from incoming and outgoing links pointing to or from the main Edinburgh article: 10,898 entries (from July 2011).
- WordNet 3.0 noun data limited to the first senses: 70,247 entries (Miller, 1995).

Across experiments, QA fired 144 search queries[1] including IM push queries and user questions. Table 1 shows statistics on the contribution of each source.

Source	#answers	#adjusted	#unique	#correct
Gazetteer of Scotland	91	89	21	89 - 100%
Wikipedia	6	6	4	6 - 100%
WordNet	18	16	9	7 - 44%
Total	115	111	34	102 - 92%

Table 1: Answer correctness per source

For 115 queries, QA found an answer from the search index. We removed those questions that should not have been sent to index search because they were anaphoric questions missed by QA (#adjusted column).
Because experiments were repetitive in nature, the table also shows the number of distinct answers retrieved overall as a better indicator of volume. Note that, in case of a tie when ranking candidates for selection, the search function always prefers the Gazetteer of Scotland as primary source, and the counts reflect that strategy. Wikipedia usually covered items out of the Gazetteer scope (e.g. *haggis*).
WordNet was a source of ambiguity and consequently errors when processing questions about Edinburgh-specific entities that could not be matched directly in the Gazetteer or Wikipedia: for instance the *Market*

[1]Note that not all QA queries fire a new search as co-refering questions and *Tell me more* types of questions make use of a cached answer.

Cross, or City Model entities pushed by the IM with an underspecified name such as *Fountain*, *Monument*. For those, WordNet provided dictionary-style definitions for the related common name. An other source of ambiguity, although not seen during evaluation, is the assignment of first senses to US entities, e.g. *Burns* first refers to a US comedian and film actor. On the other hand, when actual dictionary-like questions were asked (*What is haggis?*), answers could be found in Wikipedia. Given the high rate of false positives, WordNet has currently been removed from the current QA component.

Another feedback from subjects' experiments concerned the difficulty to understand QA snippets read out by TTS. We started looking into means of simplifying the answer text to improve TTS reading. Wikipedia entries in particular were found hard to read (longer sentences, parentheses).

To improve on this, we plugged the text provided by Simple English Wikipedia[2] for entries that were covered by the Simple English version. Editing guidelines enforce the use of basic English vocabulary and simple sentence structure (subject - verb - object). We assume these entries should be easier for TTS to read out and for subjects to listen to.

Although we linked the Simple texts wherever possible, we did keep from the main Wikipedia entry the metadata that are used for re-ranking (for instance, the number of incoming links from the main Wikipedia).

Finally, as a generic simplification sweep, we also systematically removed information in parentheses across all data sources.

Further adaptation will be needed in the area of simplification. We checked the percentage of Edinburgh-related Wikipedia entries (10,898) that were covered by the Simple English version, and only 975 were accounted for (less than 9%). However, some of these entries may prove to be among the most frequent hits. TTS also still has difficulties with UTF-8 items, such as "$\frac{3}{4}$", and abbreviations that are not always straightforward to expand automatically because of ambiguity.

4 QA features

During street evaluations, the main features of the QA component were:

- pull - user formulates a question
- push - IM pro-actively pushes answers to questions internally generated (near-by and/or visible points of interests)
- entity tracking - a limited co-reference resolver was implemented using dialogue history and IM GIS context.

The following features have been added:

- transactional queries - the IM may query QA without committing answers to the user to better plan ahead and assess in advance when and which answers should be pushed. If an answer is indeed pushed to the user, it is also committed to QA history. Committing has an impact on user pull queries that may be co-refering as the resolver will look up history to find candidates for resolution. If not committed, the question is ignored and its asking point (focus) will not be used for co-reference resolution.

[2] http://simple.wikipedia.org

- barge-in - the user may interrupt QA answers. QA is notified of the user barge-in via a callback implemented by HWU in the call handler. When a query is received, QA checked whether it has been previously talked about. If the last snippet was barged-in, it will be repeated before resuming further. Information about which asking points (foci) have been barged-in is kept for possible inference on user interests - although the motivation one a single barge-in is not clearly known: the user may barge-in because he is not interested, or because his attention may be required for navigation or other pragmatic tasks.
- topic tagging - a service that assigns a set of topics to a queried entity.
- deictic questions - QA now processes deictic questions by accessing the CM visibility engine.

The next sections review topic tagging and deictic processing in more details.

4.1 Topic Tagging

The motivation for topic tagging is to keep track of topics that have been talked about in the context of exploration. The service is used by the IM to identify user interests over time.

To capture topics of interest, we developed a corpus of sample questions and answers (Subsection 6.2.5) and annotated relevant topics. We opted for a list that is driven by the existing data, rather than an *a priori* list of possible topics relating to Edinburgh tourism. After a few annotation iterations, the following stable list of topics emerged: `people`, `architecture`, `history`, `politics`, `religion`, `literature`, `science`, `museum`, `nature`, `arts`.

Topics were captured on the basis of the first segment provided as an answer. For instance, assuming a question about the *Scott Monument*, QA replies: *The Scott Monument was build between 1840 and 1846 as a memorial to the writer Sir Walter Scott.* Topics covering this first segment would be: `architecture`, `people`, `literature`, `history`. If the user is keen on literature, he may ask as a follow-up: *Tell me more about Sir Walter Scott*. QA would answer *Prodigious writer, patriot and enthusiast for all things Scottish.* The associated topics would be `literature`, `politics`, `people`. At this stage, SPACEBOOK is informed that the user seems to have an interest in `literature` and `people`. When it comes to pushing information, SPACEBOOK can favour point of interests tagged with these specific topics.

We used machine learning to detect topics, with one learner assigned per topic. Each learner trains a language model based on the following features:

- keyword lemmata
- presence of a named entity (among: person, organisation, location, time)
- the entity is known to be human from the detected question type
- presence of verbs using the past tense

To test this set of features, we performed a 10-fold stratified cross-validation on the annotated data using Weka simple logistic algorithm. Table 2 shows the results.

Most topics are highly dependent on keyword triggers and taggers would benefit from either more training data or a list of terms relevant to the topic, obtained for instance from unsupervised clustering, to reduce annotation work.

Topic	+topic f-score	-topic f-score	Accuracy	Kappa	#examples
People	0.894	0.907	90.0%	0.8017	150
History	0.705	0.862	81.2%	0.5676	103
Architecture	0.556	0.861	78.9%	0.4229	85
Science	0.651	0.945	90.5%	0.5992	52
Politics	0.384	0.916	85.1%	0.3160	52
Literature	0.844	0.973	95.4%	0.8181	52
Religion	0.553	0.936	88.7%	0.4957	50
Nature	0.694	0.959	92.7%	0.6568	46
Arts	0.531	0.945	90.1%	0.4845	44
Museum	0.996	0.973	99.3%	0.9692	38

Table 2: Topic taggers performance. `+/- topic f-score` measures the ability of each tagger to recognise positive and negative examples. The last column indicates the number of positive examples, that is the number of segments for which the topic was assigned.

For integration in SPACEBOOK, we ran the topic taggers on the first segment of each entry in the data sources and indexed the set of discovered topics to make them readily available at runtime. When a new query is sent (whether a pull from the user or an internal push), QA returns the answer as well as the associated topics. The IM can also make independent topic queries if needed.

4.2 Deictic questions

The second major update in QA concerns co-reference resolution and more specifically deictic questions.

QA supports entity tracking by maintaining a list of the asking points (or foci) automatically identified in incoming questions (whether pulled or pushed).

When a question comes in, it is typed, the focus is extracted and marked as co-referring or not (see Deliverable 6.2.2 for these components' description and evaluation). A co-referring focus can be either anaphoric or deictic. An anaphoric focus points to an entity previously mentioned in the dialogue. It can refer to: (1) a previous entity asked about (from a question) or (2) a previous entity referred to in the QA snippet (from quoted text)[3]. A deictic focus points to an entity in the user view-shed. The goal of co-reference resolution is to track which entity is talked about.

Co-referring foci and their complements can provide hints about the sought-after entity, from simple nominal types to more elaborated constraints: *What is that **monument** on the top of the hill?* specifies a monument that is elevated and in a distance. Currently, QA uses the syntactic head of the focus as a hint for type: e.g. monument. (More elaborated constraints are not currently handled.) Table 3 summarises the different syntactic categories of co-referring foci covered by QA. The question type can be seen to play a role in entity tracking (`biography` calling for a human target and `next-segment` question types that are anaphoric questions by definition).

Disambiguating the type of focus (anaphoric versus deictic) is necessary to deliver relevant information. It is also important to make decision about QA behaviour when it comes to resuming versus delivering

[3](2) is currently not supported.

	Anaphoric	Deictic
Pronominal	User previously asked about Hume: *Tell me more about **him**.* A snippet mentions haggis, user asks: *What is **that**?*	User spots the national museum: *What is **that**?* User is in front of Hume statue: *Who is **it**? Who is **he**?*
Nominal	QA introduced Hume statue: *Tell me more about the **statue**.* QA previous talked about Hume: *Tell me more about the **philosopher**.*	User spots an interesting building: *What is this **monument**?* *What **church** is that?* In front of a statue: *Who is this **man**?*
Elliptic	*Tell me more.*	-

Table 3: Syntactic categories of co-referring foci

novel information.

If anaphoric, QA should resume information about the entity (depth-first).

If deictic, QA searches for the most salient point of interest in the view-shed that has not been talked about yet (breadth-first).

Anaphoric and deictic are mutually exclusive: if a focus is deictic, it should not refer to a previous focus.

In practice, if a user for instance repeats *What is that?*, a new point of interest will be focused on each time. We assume that this approach is more practical from a usability point of view: if the deictic target is missed, the user can ask again. It also naturally induces a one-by-one listing behaviour that is less overwhelming than providing an actual list answer (*What are interesting things around me?*) that may be tedious to listen to.

The current strategy will first check whether the focus is anaphoric, i.e. if it can be matched to an anaphoric candidate from QA history. If not, the focus will be considered deictic. The only exception concerns questions for which the focus is undefined, for instance *What is **that**?*. Because this type of focus could potentially match any anaphoric candidate, and the phrasing indicates more likely a deictic reference, for those questions, QA will first consider deictic candidates.

Deictic resolution is currently performed in two steps. First the CM visibility engine is queried with the user coordinates and heading. It returns the top N (under 10) sites and/or polygons ordered by saliency score. These candidates are then filtered to select the best match given the constraints imposed by the focus (for instance the entity type: *What is this **church**?*). QA also checks that the selected candidate has not been previously talked about. At a later stage, QA will be able to provide additional query parameters to the visibility engine instead of post-processing the result list.

There is a special handle for deictic questions typed as `biography`. For instance, if the user asks *Who is it?* in front of David Hume's statue, QA will search for a salient statue in the view-shed, extract the associated person's name (if any) and present information about the person rather than the statue. On the other hand, a question such as *What is this statue?* will be answered with information about the statue itself rather than its representation. Statues are the only points of interest that are currently potentially

transferable to a human type.
To evaluate deictic treatment, we are in the process of building a corpus (Subsection 6.2.6) that specifies user coordinates and heading, a question and acceptable points of interest corresponding to the question. The objective is to collect a variety of questions, from underspecified (*What is that?*) to more complex (*What is this big building with large columns over there?*) utterances.

5 Search component

Both component and integrated evaluations were carried out and described in D6.2.2. Table 4 is provided as a reminder of the performance in 3 experiments:

- a baseline experiment with no added re-ranking, using the default TF-IDF Lucene search
- the component evaluation on development test data (Subsection 6.2.4)
- the street evaluation

The evaluation measure is the Mean Reciprocal Rank and only the first candidate was evaluated to conform with Spacebook requirements (the user is sent a single answer, not a a result list as it may occur in multi-modal systems making use of device display).

System	Lucene	Component evaluation	Street evaluation
MMR1	0.659	0.923	0.895

Table 4: MMR1 over search experiments

The search component has been re-trained to adjust to the removal of WordNet. More questions are being added to the test corpus (Subsection 6.2.4). But the core features used for re-ranking answer candidates are currently the same and based on:

- the source of the entry (Wikipedia, Gazetteer);
- a flag if the entry was indexed as a person entity;
- the question type;
- Lucene similarity score;
- the string distance score between the focus and the main name for the entry;
- the minimum string distance score between the focus and of the possible alternative names provided by the source;
- a flag if the entry is from Wikipedia and has a disambiguation string, such *X,_London*, *X_(politician)*;
- a flag if the disambiguation string is actually Edinburgh;
- and the number of incoming links for a Wikipedia entry.

Interactive QA relies on two modules that are used for both search and co-reference resolution: question classification and focus extraction. Deliverable D6.2.2 described both component and street evaluations for these two modules. Corresponding datasets can be found in Subsections in 6.2.1 and 6.2.2. Both modules were re-trained to cover deictic questions.

6 Deliverable

6.1 Software

The software for question answering is available in the project SVN repository as a Maven project at:
`/spacebook/EDINI/spacebook-ag/trunk`
A README file contains instructions on how to build the package and access further software documentation.

A Street View development demonstration is available at:
`http://aethys.com/spacebook-qa2`
The built Maven documentation is also available at:
`http://aethys.com/spacebook-qa2/site`
(Note that, because it is a development demo, the website may be momentarily off-line and subject to changes).

6.2 QA training data

QA is mostly based on supervised machine learning techniques. This section reviews the data that has been annotated so far. Sample questions are collected from different sources:

- Wizard of Oz data,
- logged questions phrased by partners testing the QA web interface,
- logged questions coming from independent IM development (street tests as well as Google Street view tests),
- and questions generated from the rules which the initial Prolog parser was based on (in particular to collect navigation requests and train the `out-of-scope` question type).

There is only one annotator involved (Tiphaine Dalmas, UE).

6.2.1 Question types

2,134 questions were annotated with one of the following types: `biography, description, nextsegment, outofscope`.

`/spacebook/EDINI/spacebook-qa/trunk/src/main/resources/training/qtype.xml`

6.2.2 Focus extraction

Each entry corresponds to a definition question marked-up with its focus. 700 definition questions were annotated.

`/spacebook/EDINI/spacebook-qa/trunk/src/main/resources/training/focus.xml`

6.2.3 Coreference checker

Each entry corresponds to a focus context. 540 contexts were annotated as co-refering or not.

`/spacebook/EDINI/spacebook-qa/trunk/src/main/resources/training/anaphoric_entities.txt`

6.2.4 Answer candidate reranker

319 questions were annotated with correct answers from the different data sources.

`/spacebook/EDINI/spacebook-qa/trunk/src/main/resources/training/answers.definition.xml`

6.2.5 Topic tagging

Each entry lists the first segment of a given exploration question. Topics were manually assigned from the following set: `people`, `architecture`, `history`, `politics`, `religion`, `literature`, `science`, `museum`, `nature`, `arts` (10 topics). 303 entries were annotated.

`/spacebook/EDINI/spacebook-qa/trunk/src/main/resources/training/topics-v2.xml`

6.2.6 Deictic questions

This corpus is in construction. Each entry specifies a deictic question (e.g. *What is this church?*) as well as the user's coordinates and heading. It is annotated with the City Model entity that was referred to by the subject. For this corpus specifically, we are planning to involve several subjects to collect a wider variety of deictic questions.

`/spacebook/EDINI/spacebook-qa/trunk/src/main/resources/training/whatisthat.xml`

7 Conclusion

In this report, we reviewed the changes contributed to QA since the last evaluation. The main changes concerned readability for TTS, topic tagging for user interests tracking and deictic resolution. Integration to SPACEBOOK and in particular the IM and CM has been done in collaboration with HWU and Edinburgh School of GeoSciences.

References

Gittings, Bruce. 2012. *The Gazetteer for Scotland - http://www.scottish-places.info*.

Li, Xin, & Roth, Dan. 2002. Learning question classifiers. *In: 19th International Conference on Computational linguistics*.

Mikhailian, Alexander, Dalmas, Tiphaine, & Pinchuk, Rani. 2009. Learning foci for Question Answering over Topic Maps. *Pages 325–328 of: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*.

Miller, George A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, **Vol. 38, No. 11**, 39–41.