Natural Language Engineering **19** (1): 1–32. © Cambridge University Press 2012
doi:10.1017/S1351324911000350

When the answer comes into question in question-answering: survey and open issues

ANA CRISTINA MENDES and LUÍSA COHEUR

Spoken Language Systems Laboratory - L²F/INESC-ID
Instituto Superior Técnico, Technical University of Lisbon
R. Alves Redol, 9 - 2º– 1000-029 Lisboa, Portugal
e-mail: {ana.mendes, luisa.coheur}@l2f.inesc-id.pt

(*Received 24 September 2011; revised 4 October 2011; accepted 14 November 2011; first published online 10 January 2012*)

Abstract

The answer determines the success of a Question-Answering (QA) system. In redundancy-based QA systems, a common approach is to extract the candidate answers from the information sources and select the most frequent answers as the final answers. However, this strategy has some pitfalls. For instance, if a system is not able to detect equivalences between the candidate answers, their frequencies might be erroneously calculated. Moreover, the user who posed the question should also be taken into account when answering: different persons require different (correct) answers. This can involve the use of suitable vocabulary and/or information details. In these situations, the generation of a response can be a more suitable strategy, instead of the extraction and direct retrieval of the answer from the information sources. The present survey targets the state of the art in the answering task in QA under three different lines of research. First, we present several works that focus on relating candidate answers. Then, we recover the concept of cooperative answer – a correct, useful, and non-misleading answer – and we bring up attempts to address cooperative answering. Finally, we investigate the research community endeavors on response generation. We will also present our perspective on each of these three topics throughout this paper.

1 Introduction

Question-Answering (QA) systems automatically answer questions posed in natural language. To this end, a common approach to solve open-domain factoid questions relies on the direct extraction of candidate answers from document collections or the web, while the selection of the final answer(s) is considered a ranking problem that benefits from the redundancy of the used information sources. A score is attributed to each candidate, usually as a function of its frequency of occurrence, and the final answer(s) is/are the top N candidates selected from a list sorted in decreasing order of score, presented afterwards to the user.[1] In this survey, we will refer to the

[1] The architecture of a QA system has become quite standardized: It is a pipeline composed by three main components dedicated to the analysis of the question, the retrieval of documents/passages from the information sources, and the extraction and selection of

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

aggregated tasks of selecting and presenting the final answer(s) to the user as *the answering task*.

A common step in redundancy-based QA systems is the detection of relations of equivalence between candidate answers, i.e., that some answers represent the same entity despite their surface form: for instance, *Oct. 14, 1947* is the same as *14th October, 1947*. Some QA systems identify clusters of equivalent candidate answers; however, researchers and developers should be aware that other semantic relations exist between candidate answers that can improve the answer selection task. In fact, some works already take these relations into consideration; for instance, Dalmas and Webber (2007) detect relations of equivalence and inclusion in order to choose the final answer. A different perspective on the importance of relating candidate answers is presented by Buchholz and Daelemans (2001), who identified several situations where answers can be combined through the relations established between them to build a more user-oriented answer. Thus, the task of *relating candidate answers* arises as a way to improve the performance of QA systems by aiding the selection and presentation of the final answer(s).

When it comes to choosing the final answer, the trend in QA is to consider the retrieval of correct answers as the ultimate goal. This view is promoted by the evaluation guidelines of QA systems that uniquely regard (and reward) correct answers as valid answers to the posed questions. Whereas this practice is sufficient for benchmarking purposes, allowing a straight comparison between systems, it might not be enough when the answer is to be given to a real user. In this context, according to Gaasterland, Godfrey and Minker (1992) better than a correct answer is a cooperative answer, that is, a correct, non-misleading, and useful answer. The concept of cooperative answering (Kaplan 1979) appeared in the late 1970s in response to a known, yet undesirable, property of databases – they will always retrieve the literal answer to a query even if it misleads the user – and has been extensively studied in the context of answering database queries. However, despite the fact that QA has moved its paradigm from structured to free text data, the need remains to respond to the user with a cooperative answer. *Targeting cooperative answers* appears as a way to attribute a more human-like behavior to the system. The YourQA system (Quarteroni and Manandhar 2007) took the first steps in this direction by adapting its answers to the user's previously defined profile.

The job of a QA system is not complete until it answers the posed question. As previously mentioned, a common strategy is to directly retrieve the top N scored candidate answers extracted from the information sources. Superficially, this might be considered sufficient. However, it ignores other relevant information that can contribute to the system's response. For instance, the degree of confidence of the system in the given answer: If the system is not totally confident in the answer, a response such as *I believe the answer to your question is...* would be more appropriate than the answer alone. Also, note the importance of response generation for the purpose of showing cooperativeness: the user will not be misled by an answer that

the answers. Readers can get details about this architecture, for instance, in Jurafsky and Martin (2008).

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

the system is not sure of being correct. Thus, in the pipeline of a QA system, the last desirable step is to *generate the response*, building it from the answer the system believes to be correct. The work of Moriceau (2005) is an example, for she developed an approach that lexicalizes and computes the response to be returned to the user.

In this paper, we survey the state of the art in the answering task in QA and adopt the concept of cooperative answer as presented previously. This document is composed of three main sections: relating candidate answers, targeting cooperative answers, and generating responses. Since there are many loose ends concerning these topics, whenever possible we will make a bridge with other areas that face similar problems and from which some techniques can be borrowed.

This paper is organized as follows: Section 2 focuses on relating candidate answers, presenting a general typology of relations, and describing techniques for their detection; Section 3 addresses the need of targeting cooperative answers; and Section 4 focuses on techniques and systems that generate the system's response. The paper finishes in Section 5, where conclusions are drawn and we discuss open issues and point to future directions in QA.

2 Relating candidate answers

In the early 2000s, QA saw a change of paradigm from knowledge- to data-driven with the works of Brill *et al.* (2001), Clarke, Cormack and Lynam (2001), or Kwok, Etzioni and Weld (2001), even if each one explored a different strategy. What was once considered an effortful, linguistic, and domain-dependent task, proved its success under conditions where knowledge is not explicitly stated if a large amount of data are provided. The idea is to save deep computational processing by taking advantage of the statistical properties of large-scale information sources. The advent of the web favored this change, allowing resource orders of magnitude larger than any other built document collection to become publicly available, accessible to everyone, and from everywhere.

The redundancy-based approach in QA has proven to achieve good results, while requiring little effort. It relies on the premise that it is probable that the answer to a question is stated several times, in multiple forms, in the information sources. An example of the application of this approach is the Aranea system (Lin 2007), a re-implementation of the AskMSR system by Brill *et al.* (2001). Aranea extracts n-grams from the information sources and after voting, filtering, and ranking, chooses the better scored as the final answer. This rather simple technique attained a precision of 47.7% for nearly 1,600 questions gathered from several QA tracks of the Text REtrieval Conference (TREC).

The redundancy-based strategy usually considers candidate answers as independent entities. However, this is often not the case. For example, the candidate answers *Milan* and *Italy* to the question *Where is the famous La Scala opera house?*[2] are related (*Milan* is located in *Italy*) and probably they should not be dealt with

[2] The majority of the examples in this survey are taken from Magnini *et al.* (2003), Magnini *et al.* (2005), and Moriceau (2005).

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

separately. Detecting and using the relations between candidate answers can improve the performance of QA systems by addressing a threefold purpose: (1) allowing answer comparison and integration from various and heterogeneous sources; (2) improving answer selection (Dalmas and Webber 2007); and (3) allowing to build *better* answers (Buchholz and Daelemans 2001).

In this section, we describe a general typology of semantic relations between answers, originally introduced by Webber, Gardent and Bos (2002) and later presented by Moriceau (2005).[3] Note, however, that the two previous works consider relations between correct answers, assuming a filtering phase where the incorrect ones are discarded. Here, we discuss the relations between candidate answers and how they can influence the computation of a correct answer. We also present techniques for their detection.

2.1 A typology of relations

According to Webber *et al.* (2002) and Moriceau (2007), relations between two candidate answers can be of following types:

Equivalence – If answers are consistent and entail mutually. Equivalence exists between lexicographically different answers that represent the same entity, namely,

(1) answers with notational variations. For instance, *Oct. 14, 1947* and *14th October, 1947* are equivalent answers for *When did the test pilot Chuck Yeager break the sound barrier?*;
(2) answers that rephrase others, like synonyms or paraphrases. For example, the question *How did Jack Unterweger die?* can be answered with *committed suicide* or *killed himself*.

Inclusion – If answers are consistent and differ in specificity, one entailing the other. Inclusion occurs between two candidates that represent different entities, in which one includes or subsumes the other, through

(1) hypernymy, that is, the answers are in a *is-a* relation. For example, *What animal is Kermit?* can be answered with *frog* or *amphibian*;
(2) meronymy, that is, the answers are in a *part-of* relation. For example, *Where is the famous La Scala opera house?*, in which *Milan*, *Italy*, *Europe* and *Earth* are the possible answers;
(3) membership, that is, the answers are in a *instance-of* relation. For example, *Edvard Munch* is a member of *a Norwegian Symbolist painter*, both being possible answers to *Who painted the 'Scream'?*.

Aggregation – If answers are consistent, but not mutually entailing. In aggregation, all candidate answers are potentially correct and can be integrated in the form of a conjunction. For example, the question *Where is the Disneyland?* can be

[3] In her PhD thesis, Moriceau (2007) also mentions the relation of complementarity existing between candidate answers that require inference mechanisms to solve a given question.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

Table 1. *Relations between candidate answers*

Moriceau (2005)	Webber *et al.* (2002)	Buchholz and Daelemans (2001)
Equivalence	Answers determined to be equivalent (mutually entailing)	Different measures Different designations Time dependency
Inclusion	Answers that differ in specificity (one-way entailing)	Granularity
Aggregation	Answers that are mutually consistent but not entailing	Collective answers
Alternative	Answers that are inconsistent	Many answers Ambiguity in the question Different beliefs

answered with *Paris*, *Tokyo*, *Hong-Kong*, or *Los Angeles*, or a conjunction of all: *Paris, Tokyo, Hong-Kong, and Los Angeles.*

Alternative – If answers are not consistent. In the case of questions with a single answer, only one of them can be correct. For example, the question *How is the Pope?* can be answered with *ill* or *healthy*, but not with both. In the case of questions with multiple answers, all the alternatives may be distinct answers. For example, *twenty-eight* and *twenty-nine* are alternative answers to the question *How many days are in February?*.

Table 1 shows the existing relations between candidate answers. A parallel is also made with the work by Buchholz and Daelemans (2001), who focused on relations in the context of the presentation of complex answers (answers composed of two or more simple answers), and as presented by Dalmas and Webber (2007).

2.2 Techniques for relating answers

In the following, we discuss how to detect relations between answers and the techniques to be used. We point to actual research on this topic.

Relations of equivalence: A common task before the detection of relations of equivalence is the normalization of answers, that is, the reduction of spelling variations of each answer/entity to its canonical unambiguous referent. The task of normalization is often used to allow comparison and integration of candidate answers despite their surface form. Normalization can vary from simple rewritings to the creation of complex objects containing detailed information that characterizes the answer. The JAVELIN system at TREC 2002 (Nyberg *et al.* 2002), for instance, converts candidate answers to a canonical form depending on their semantic category: answers of category LOCATION are reduced to the short form specified in the CIA World Factbook;[4] answers of category NUMERIC to a unit-value pair; answers of category DATE

[4] Available at `https://www.cia.gov/library/publications/the-world-factbook/` Last accessed in November 2011.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

to the format *mm/dd/yyyy*; and answers of category PERSON-NAME are split into first and last names and are lowercased. Moriceau (2005, 2006) uses normalization to compare and integrate answers of categories DATE and NUMBER. Answers of category DATE are assumed to be either temporal points, e.g., *September 16th, 1989*, or time intervals, e.g., *September 1989, from 10 to 22*, both being normalized to the form of intervals [*16-9-1989,16-9-1989*] and [*10-9-1989,22-9-1989*], respectively. Concerning the answers of category NUMBER, each is represented by a frame that contains all its relevant properties such as its unit of measure, precision, and numerical value.

Equivalence is usually seen as orthographic similarity. Therefore, one way to detect the relation of equivalence consists of measuring the string similarity between candidate answers. This can be done with recourse to metrics such as the cosine or Jaccard similarities, or the Levenshtein distance. If the similarity (distance) measured between two candidate answers is higher (lower) than a given threshold, then they are similar enough to be considered equivalent. For instance, Téllez-Valero *et al.* (2010) use the Levenshtein distance in order to cope with language variability and typing errors in answers from different sources.

Another way to detect equivalence is based on the information overlap between the candidate answers. If the number of tokens shared by two answers is higher than a given value, then they are considered equivalent. For instance, the Mulder system (Kwok *et al.* 2001) uses the answers' overlap to cluster similar answers – *Alan Shepard*, *Shepard*, and *Alan B. Shepard* are assigned to the same cluster – and the AskMSR system (Brill, Dumais and Banko 2002) creates longer answers by merging shorter similar answers – *San*, *Francisco* and *San Francisco* are aggregated into a unique answer: *San Francisco*.

Similarity-based techniques are known for their simplicity; however, it is clear that they have some limitations: for instance, the equivalence between *George Bush* and *George W. Bush* can not be detected in the same way as the equivalence between *John Kennedy* and *John F. Kennedy*. In the former case, context is needed to decide whether they refer to the same entities or not.

Other works take equivalence to a semantic level. For instance, the Pronto QA system (Bos, Curran and Guzzetti 2007), built around the Discourse Representation Theory (Kamp and Reyle 1993), expresses the meaning of each question and the respective candidate answers through discourse representation structures, which allow semantic comparisons between the answers. Dalmas and Webber (2005) use WordNet's synonyms (Fellbaum 1998)[5] to test whether two answers share the same meaning: if so, they are considered equivalent. They also perform orthographic and lexical comparisons of the answers, comparing their lemmas/stems, recognizing abbreviations, and calculating the Levenshtein distance between them.

[5] Available at `http://wordnet.princeton.edu/` Last accessed in November 2011.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

As done by Dalmas and Webber (2005), the detection of the relation of equivalence often relies on external knowledge repositories. Another example is the framework of Ko, Si and Nyberg (2007) for answer selection in QA that uses lists of synonyms collected from three knowledge bases – WordNet, Wikipedia,[6] and the CIA World Factbook – together with several string distance metrics and a set of handcrafted rules to compute the similarity (equivalence) scores of candidate answers.

Relations of inclusion: The lexical relations in the WordNet are also used to detect inclusion between candidate answers. Dalmas and Webber (2007) recognize inclusion if two answers share the same lexical head (this is done through string matching) and if they are connected through a *hypernymy/hyponymy*, a *part of* or a *member of* pointer in the WordNet; hence, *western Pacific* is recognized as being included in *Pacific*.

Although, to our best knowledge, the work by Dalmas and Webber (2007) is unique in detecting and dealing with inclusion between candidate answers, several works in QA have already benefitted from the existence of this relation in a variety of tasks. In answer extraction, the correct answer to a question is commonly a hyponym of the target of the question, for example, the correct answer to *What is the largest variety of cactus?* is one of the hyponyms of *cactus*. In particular, Paşca and Harabagiu (2001) have suggested that the members of the WordNet classes can be candidate answers to a question. In the classification of answers, the category of the answer's hypernym can be used when the category of a candidate answer cannot be found (Nii *et al.* 2004). In the classification of questions, the hypernym of the question headword is often used as a feature to feed a machine learning-based classifier (Huang, Thint and Qin 2008; Silva *et al.* 2011).

The detection of semantic relations between words, such as the inclusion, has been widely explored in the task of *knowledge acquisition* where the goal is to acquire relations and their arguments from text. An example is the seminal work by Hearst (1992), where a bootstrap algorithm was proposed to identify patterns to extract hypernym/hyponym pairs from large text corpora based in a set of seeds. Hearst (1992) was followed by several others (Pantel and Ravichandran 2004; Ritter, Soderland and Etzioni 2009). Although applied to a different task, the employed techniques can be brought to QA.

Relations of aggregation and alternative: When two candidate answers A_1 and A_2 are alternatives, then the information they convey is in contradiction. Transposing to logic, there is no world in which A_1 and A_2 are both true: the Pope cannot be *ill* and *healthy* at the same time. That condition does not hold in the relation of aggregation, since here it is necessary that at least one world exists in which A_1 and A_2 are both true. Particularly, in QA the difference between aggregation and alternative is dependent on the domain under consideration. For instance, consider the question *What is Vertigo?* and the candidate answers *an illusion of movement* and *the name of a movie*. In

[6] Available at `http://wikipedia.org/` Last accessed in November 2011.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

open domain QA both candidate answers are in aggregation: *Vertigo is an illusion of movement and the name of a movie*; however, in the medical domain, the second should be considered incorrect. Here, the candidate answers are in a relation of alternative where only *an illusion of movement* is correct.

According to Moriceau (2007), in a total of 203 relations existing between the correct answers to 108 questions, 25% were of equivalence, 8% of inclusion, 28% of aggregation, and 35% of alternative,[7] the last two being the most common relations. Nevertheless, to our best knowledge, no work in QA has yet focused on the actual detection of alternative and aggregation. Instead, QA systems tend to assume that all nonequivalent answers are alternative to each other: this is, once again, the view that sees each candidate as a competitor (Dalmas and Webber 2007).

Although the detection of relation of alternative between candidate answers has not been fully explored in QA, the detection of contradiction was already considered in the Natural Language Processing (NLP) field. Two known research works that deal with contradiction are those of Harabagiu, Hickl and Lacatusu (2006) and De Marneffe, Rafferty and Manning (2008).

Harabagiu *et al.* (2006) specifically focused on contradiction introduced by antonymy, negation, and contrast.[8] These authors train a maximum entropy classifier to indicate the existence (or not) of a contradiction in a pair sentence/hypothesis and report results of 75.63% of precision for the cases when the hypothesis is a negated version of the sentence, and 62.55% when the hypothesis is a negated paraphrase.

De Marneffe *et al.* (2008) deal with a broader range of contradictions and adopt a loose definition of this linguistic phenomena: 'contradiction occurs when two sentences are extremely unlikely to be true simultaneously'. These authors first introduce a typology of contradictions, distinguishing those that arise from antonymy, negation, and number/date mismatch, from those that are due to the use of factive or modal words, structural and lexical contrasts, and world knowledge. Contradiction between sentences is detected by means of logistic regression that manipulates several contradiction features extracted from both sentences; for instance, antonymy-related features are based on lists of antonyms from the WordNet and the VerbOcean[9] (Chklovski and Pantel 2004). Good performance on contradiction arising from negation and antonymy is reported (with a precision of 62.74% and 78.9%, respectively). However, the authors indicate that several difficulties were found when the detection of contradiction requires lexical and world knowledge.

[7] The remaining 16% were left to the complementarity relation.

[8] As defined by Kehler (2002), contrast is a resemblance discourse relation that can be recognized by the presence of the phrases 'but' or 'although'.

[9] Available at `http://demo.patrickpantel.com/demos/verbocean/` Last accessed in November 2011.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

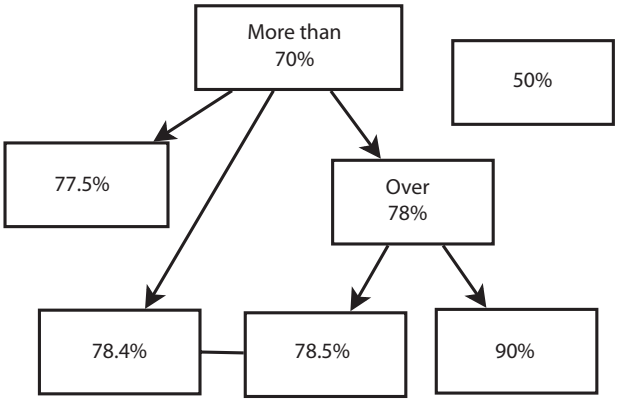


Fig. 1. Graph of numeric candidate answers related by semantic relations.

2.3 Discussion

The focus of the current section is on the semantic relations between answers and the techniques employed to detect them. Our view is that these relations have a positive impact on the task of choosing the correct answer from a list of candidate answers, and we have conducted a preliminary experiment to test this hypothesis (Mendes and Coheur in press).

We gathered the answers given by the competing systems in the QA tracks of TREC from 2003 to 2007, judged as correct or incorrect by human assessors. Since the answers are originated in actual systems that used the same information sources to solve the same set of questions, and their evaluation is trustworthy, we consider this a reliable and consistent corpus that allows to mimic the behavior of a real system, which collects information from various and heterogeneous sources.

The relations of equivalence and inclusion between candidate answers are detected by means of hand-built rules and by using the lexical relations in the WordNet. Afterwards, we built a directed graph of candidate answers in which the answers are the nodes and relations between edges. This approach is similar to that of Dalmas and Webber (2007); however, we use answers provided by real systems and focus on a broader variety of answer categories: DATE, ENTITY, HUMAN, and NUMERIC (from Li and Roth 2002). An example of such a graph can be seen in Figure 1. Straight lines represent relations of equivalence, and arrows represent relations of inclusion, where the answer at the start of the arrow includes the answer at the end of the arrow.[10]

The score of an answer *A* is updated according to its frequency, the number of answers equivalent to *A*, the number of answers that include *A*, and/or the number of answers that are included by *A*. This strategy boils down to navigating the graph of candidate answers and adding to the score of each node a weight that depends on the number and types of edges that leave from and arrive to it. Finally, a list of candidate answers is built from the graph, ordered by decreasing value of score

[10] For simplicity reasons, we omitted transitive inclusions.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

Table 2. *Accuracy results of a preliminary experiment to assess the impact of semantic relations in the selection of final answer*

Category	Frequency (Baseline)	Normalization + frequency	Normalization + frequency + equivalence and inclusion
HUMAN	44.92%	44.92%	52.01%
ENTITY	39.29%	40.18%	42.41%
NUMERIC	28.53%	32.07%	44.02%
DATE	46.58%	53.42%	57.98%
Overall	39.78% (526/1322)	42.61% (562/1322)	49.55% (655/1322)

(equally scored answers are positioned at the same rank) and we assume that any decision about picking answers from the list in a certain rank is uninformed, i.e., it is random (uniformly distributed).

Being so, for a question q in the test set, we measured the probability of choosing the correct answer at a rank r as $P(CorrectA_{r,q}) = \frac{\#CorrectCA_{r,q}}{\#CA_{r,q}}$, where $\#CorrectCA_{r,q}$ is the number of correct candidate answers and $\#CA_{r,q}$ is the number of candidate answers at that rank. The accuracy is defined as the total questions for which the probability of choosing the correct answer is 1.0 at rank 1 divided by the total number of questions in the test set.

Table 2 shows the comparison between the results achieved when the score of a candidate answer equals its frequency (the baseline), when we apply normalization to all candidate answers before scoring, and when the relations of equivalence and inclusion are used to compute the score of each candidate.

As can be seen, when the relations between candidate answers are set and used by the scoring system, the results increase nearly 10% and 7% when compared to using only frequency and normalization plus frequency, respectively.

As an example of a correct answer pushed to the top rank, *sheep* was selected as a final answer to a question whose candidates were related as follows: '*animal* `includes` *sheep*' and '*sheep Dolly* `equivalent to` *sheep*'.

From our analysis of the results, we could see that the approach that takes into account the semantic relations better groups the correct answers in the top positions of the list of scored candidate answers. The largest difference in the number of correct answers was achieved from the category NUMERIC, where a total of four rules for detecting equivalence and eight for detecting inclusion resulted in forty-four more correct questions at the top rank when compared with the results of frequency after normalization. In the category ENTITY, this difference was the smallest. We believe that this happened because ENTITY is a broad category that covers very diverse questions and probably other techniques are required.

The achieved results strengthen the applicability of semantic relations in the task of answer selection for a large variety of questions.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

Table 3. *Database stonewalling in a user-system interaction*

User: *Which students got a grade of F in CIS500 in Spring '77?* System: *NIL*
U: *Did anyone fail CIS500 in Spring '77?* S: *No*
U: *How many people passed CIS500 in Spring '77?* S: *Zero*
U: *Was CIS500 given in Spring '77?* S: *No*

3 Targeting *cooperative* answers

Question-Answering finds its roots in research in databases, as the first systems were natural language front-ends to databases (Androutsopoulos, Ritchie and Thanisch 1995). A known, yet undesirable, property of databases is that they will always answer *yes* or *no* to a *yes/no* query, regardless of that being a misleading answer or not. Kaplan (1979, 1981) coined this behavior as 'database stonewalling' and recognized the need to give the user a more appropriate response, rather than the literal answer directly retrieved from the database. The typical example to illustrate this situation is the interaction shown in Table 3 (from Kaplan 1979).

Certainly a human answerer would immediately inform the questioner that *CIS500 was not given in Spring '77*. This and other similar examples pointed out the need for information systems to emulate the human cooperative behavior in conversations. Particularly, cooperative answering arose as a way to face such undesirable behaviors in interaction with databases, inspired by the four maxims that characterize human behavior (Grice 1975): *quality*, *quantity*, *relation*, and *manner*.

Much work was devoted to this topic since Kaplan's PhD thesis (1979), thoroughly described in the overview of Gaasterland *et al.* (1992) on the foundational work in cooperative answering in natural language interfaces and databases. According to these authors, a cooperative answer conveys the properties of being correct, non-misleading, and useful. In this survey, these three properties are considered as follows: an answer is correct if it solves the input question; an answer is non-misleading if it avoids misinterpretations; and an answer is useful if it is relevant to the user who posed the question.

Finding the cooperative answer introduces a higher level of complexity for systems to deal with, since it is not enough for an answer to be correct. Moreover, the way the properties relate is far from trivial. For instance, a correct answer might not be useful, and a useful answer might not be correct. As an example, consider the question: *Where is the famous La Scala opera house?*. The answers *Europe*, *Italy*, *Milan*, and *45º 28' 4" N, 9º 11' 19" E* are correct, but while the first one is probably not useful, the last one might only be useful for a (lost) tourist with a GPS. By the same token, for an IT student the useful answer to the question, *How many kilobytes are there in a gigabyte?* would be the correct one: *1,048,576 KB*; however, for someone who wants to have an idea of the magnitude of that amount, the useful

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

answer could be simply *1,000,000 KB*, which is not correct. Also, a correct answer can be misleading. Just take a look at the three first answers of the interaction in Table 3.

With the change of focus from structured to unstructured data, and from databases to free text, new challenges are posed to systems and alternatives to cooperative answering are to be found. In this section, we research studies that take efforts to bring cooperative answering to open domain QA. Each section covers advances in QA regarding the above-mentioned three properties of a cooperative answer, i.e., being correct, non-misleading, and useful.

3.1 Correct answers

The definition of correct answer is usually consensual: an answer is correct if it solves the input question. However, the criteria used to evaluate the results of a system are often more stringent. For example, in the QA tracks of the Cross Language Evaluation Forum (CLEF) and TREC, the output of a system is evaluated as *correct* (Dang, Kelly and Lin 2007; Forner *et al.* 2009) if

- it contains the answer to the question; otherwise it is considered *incorrect*;
- it is exact and contains neither more nor less information than required; otherwise it is considered *inexact*;
- it contains a supporting snippet of text; otherwise it is considered *unsupported.*

Returning the correct answer to a question is not a trivial task and certainly it is not completely solved by state of the art QA systems: the best performing QA system at TREC 2007 achieved a first-answer accuracy of 70.6% in factoid questions (Dang *et al.* 2007).

Recalling the generic architecture of the factoid open-domain QA system (Jurafsky and Martin 2008), the task of choosing the final answer(s) from a list of candidate answers is allocated to the answer selection component. In the following, we describe several techniques for answer selection, grouped in three major categories: (1) filtering of candidate answers, (2) ranking of candidate answers, and (3) validation of candidate answers.

The first group of techniques to answer selection implies filtering out the candidate answers that do not agree with certain constraints, typically related with the linguistic properties of the expected answer, namely,[11]

Lexical properties: The Aranea system (Lin 2007) uses a set of heuristic rules to filter out n-grams that are 'obviously wrong' answers; for instance, to questions that start with the phrase *How far* or *How tall* only n-grams with a numerical component are allowed; to questions that start with *Who* or *Where* a capitalized n-gram is required. All the other candidate answers are rejected;

[11] Here we focus on answer selection by filtering the candidate answers extracted from the information sources. Nevertheless, readers should be aware that several QA systems prefer to directly extract candidate answers that match the constraints, thus avoiding this filtering process, for instance Hermjakob (2001) and Mollá and Gardiner (2004).

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

Other constraints imposed by the Aranea system at the lexical level are related to the information overlap between the posed question and candidate answer: if the latter is contained in the former, it is also discarded.

Syntactic properties: The system of Cardie *et al.* (2000) only considered 'noun phrases' as candidate answers, while all the other syntactic constituents were excluded from being final answers, which lead to a slight improvement on the achieved results.

Semantic properties: Schlobach *et al.* (2007) present an extensive work on semantic filtering of candidate answers by using different knowledge- and redundancy-based methods. Filtering based on semantics usually presupposes a phase where the question is annotated with information about the expected answer type. For instance, a question that starts with *In which city* expects a city as answer and should be marked with a label LOCATION:CITY,[12] or some equivalent. Indeed, *question classification* – mapping the question into a predefined category that represents the type of the information expected to be in the final answer – is by no means an unquestionable step in the generic architecture of QA systems (Jurafsky and Martin 2008)[13] and a task transversal to many systems.

The second group of techniques to answer selection relies on ranking candidate answers. This is done by attributing to each candidate a score that depends on its relevance to the given question. The best scored candidate answer(s) are ranked higher in the list of candidate answers and considered more likely to be the correct answer(s) to the question.

The score is calculated by taking into account several features extracted from the posed question, the candidate answer and, eventually, the sentence where it was found. Examples of features are the frequency of the candidate answer, the information overlap between the candidate answer and the question, and the rank of the sentence in the total relevant sentences. Some systems, such as the one presented by Moldovan *et al.* (2001), heuristically combine these features, attributing a manually tuned weight to each of them. Other systems adopt machine learning-based strategies to automatically learn the feature weights. For instance, Ittycheriah, Franz and Roukos (2001) used a maximum entropy model to combine features and their weights; the features include query expansion features, focus features, named entity features and dependency relation features.

The third group of techniques to answer selection is based on answer validation. A definition of answer validation comes from Harabagiu and Maiorano (1999), who see it as an inference process that considers an answer to a question to be valid if an explanation for it can be found. Moldovan *et al.* (2003) introduced COGEX, a logic prover for QA that aims at understanding the relations between the question and each candidate answer through the use of linguistic and world knowledge axioms, and uses logical inference to validate the candidate answers. Validating answers with the COGEX leads to an increase in performance by 30% on achieved results.

[12] Here we used the question-type taxonomy of Li and Roth (2002).

[13] The impact of question classification in QA has been addressed in the study of Moldovan et al. (2003).

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

In this perspective, answer validation is a highly knowledge-dependent task. However, for the open-domain QA systems that search for answers in large-scale information sources, knowledge intensive strategies for answer validation are deemed impracticable due to the need of intensive computational processing. A solution to tackle this problem is again to take advantage of the statistical properties of large amounts of data. For instance, Magnini *et al.* (2002) see answer validation from a data-driven point of view and assume that a candidate answer to a question is valid if question and answer often co-occur. First, several keywords are extracted from the answer and question and, with those, a reformulation pattern is created, which is afterwards submitted to the web. The answer validity score is calculated based on the number of retrieved documents.

Another example is the work of Ko *et al.* (2007), who formalize the task of answer validation as the estimation of the probability $P(correct(A_i)|Q, A_1, \ldots, A_n)$ of a candidate answer A_i being correct given the posed question Q and the similarity between A_i and the other candidate answers. These authors use logistic regression and four types of features (knowledge-based, data-driven, string distances, and synonyms) and report more than 100% of average improvement in results to 1,760 questions collected from TREC when compared to the baseline where the top scored answer is retrieved.

Answer validation is related with the system's ability to self-judge the answer it is about to present to the user. The importance of answer validation in a QA system motivated the creation of the Answer Validation Exercise (AVE) (Rodrigo, Peñas and Verdejo 2009) in CLEF. The AVE exhorts systems to automatically verify the correctness of an answer to a question, given a snippet that supports it. Most participating systems follow an approach based on textual entailment, where a hypothesis is built with the question and answer, and afterwards tested against the snippet.[14] In order to decide about the correctness of the answer, the competing systems use a large variety of approaches that can be roughly split into the following two categories:

- Based on lexico-syntactic information – relying on the analysis of syntactic dependencies and/or overlapping of lexical or syntactic information between the text and the hypothesis. For instance, Iftene and Balahur (2008) compute a fitness value based on the matching between the named entities on the hypothesis and the entities of the supporting text.
- Based on machine learning – using a training corpus and a set of features, such as the Levenshtein distance or the size of the longest common subsequence between the hypothesis and the supporting snippet (Kozareva, Vazquez and Montoyo 2006; Téllez-Valero *et al.* 2008).

Selecting the final answer(s) boils down to distinguishing between correct and incorrect candidate answers. In this perspective, answer selection can be cast to a binary classification problem, where the system classifies each candidate answer as

[14] In 2006, the hypothesis was given instead of the pair question/answer. In the following years, each system was given a triple (question, answer, snippet).

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

positive (correct) or negative (incorrect). If answer selection is seen as a filtering task, the candidate answers that do not agree with the constraints are the negative examples; in the candidate answer ranking, the classification depends on the answers' score against a threshold value: if the score is greater than the threshold, the candidate answer is a positive example, classified as correct; in the candidate answer validation, the positive examples are the candidate answers that can be verified and justified by the system.

3.2 Non-misleading answers

A non-misleading answer prevents the user to create a wrong interpretation about the topic under consideration. Here we focus in three major types of answers that can mislead the user when interacting with the open-domain factoid QA systems: (1) `NIL` answers, (2) ambiguous answers, and (3) answers with erroneous granularity.

3.2.1 `NIL` answers

Correct answers are greatly rewarded in evaluation campaigns and although it is often considered that 'no answer is better than a wrong answer', usually there is no distinction in scoring between returning a wrong answer to a question and returning nothing. Indeed, albeit a wrong answer totally contradicts the goal of QA (note that to be able to identify a wrong answer is still an open research problem), returning 'no answer' is not better, since it does not bring any valuable information to the user and can lead him/her to misinterpretations. If one is to build a cooperative system, what to present when no answer was found is also an issue.

On one hand, questions with 'no answer' can be originated on the system side if it cannot find the answer in the available information sources. The common approach to circumvent this problem consists of relaxing certain constraints imposed by the query used to access the information sources. For example, the Falcon system (Harabagiu *et al.* 2000) iteratively refines the query in three different loops, where morphological, lexical, and semantic variations of certain keywords in the question are added, while others are dropped; the JAVELIN system (Nyberg *et al.* 2003) sets different constraints in the query in every relaxation cycle, for instance, low relevance keywords are discarded and a word proximity window is adjusted (that is, the size of the window where the question keywords and data types are assumed to be in a document relevant to the question).

On the other hand, questions with 'no answer' can be originated on the user side. Many questions simply have no answer, in particular those with misconceptions. Consider the question *Who is the King of France?*. Knowing that France is a republic, answering *no one* or `NIL` can make the user think the system was unable to find the correct answer. Therefore, an explanation is due showing why no answer exists to that question. Detecting and correcting misconceptions in the user's question was a subject of much research since the first works from Kaplan (1979, 1981), and extensively presented in the overview of Gaasterland *et al.* (1992). A more

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

recent system, the WEBCOOP system,[15] a logic-based cooperative QA system in the tourism domain, integrates knowledge representation and reasoning to detect misconceptions in questions and deliver non-misleading answers (Benamara 2004b). When facing a misconception, the WEBCOOP system presents it to the user with alternative solutions. To our knowledge, this problem is not dealt with in the open-domain QA, mostly due to the difficulty of the task without appropriate knowledge.

3.2.2 *Ambiguous answers*

Natural language is inherently ambiguous and QA systems are mindful to this problem. Ambiguity, as it is commonly considered in QA, arises either from corpora sources or users' questions. Systems that try to cope with the problem of ambiguity in the user's question usually push its resolution to the user side. A common strategy is to interact with the user through *clarification dialogues* in order to refine the posed question (see, for instance, De Boni and Manandhar 2003). However, ambiguity in answers is usually not addressed. Returning an ambiguous answer to an unambiguous question is not cooperative, since it leaves room to multiple interpretations instead of being a clear statement of the meaning of the answer. Consider the next question and a possible answer to it:

Q: *Who was the President of the US during the Gulf War?*
A: *President Bush.*

The ambiguity exists since there were two presidents of the United States called Bush and the answer does not clearly define which one responds to the question. Another interesting example is the following, where both question and answer are ambiguous:

Q: *Who was the President of the US during the War?*
A: *President Bush.*

Whereas no major problem seems to accrue from the previous example in the current evaluation frameworks of QA systems, this is not an appropriate behavior of a QA system in user–machine interactions. The system is assuming a certain information need from the user, in this case that he/she is referring to the Gulf War, which might not be real. That is the case of QA systems operating in *e-learning* scenarios. In this context, Hung *et al.* (2005) describe their hybrid method to sense disambiguation of question/answer pairs (stored in a knowledge base) and the posed question based on the WordNet synsets and on the number of retrieved results by a web search engine. Although the main goal was to overcome the need of students to pose their questions using the same word forms as they exist in the set of questions previously stored, this work is relevant in the sense that each answer is also semantically disambiguated (even if together with its question). Despite this effort, research is still lean on this problem.

[15] Although this survey addresses the open-domain factoid QA, the WEBCOOP system is an example of the restricted-domain QA system.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

3.2.3 The granularity problem

Whereas ambiguity plays a role of great interest to researchers and developers mainly on the question side, the problem of wrong granularity is usually observed on the answer side. Consider the following two examples:

Q: *How many people live in London?*
A: *more than 5 thousand.*

Q: *How many people live in London?*
A: *more than 5 million.*

The problem here is not related with the correctness of the answers (since both are correct), but with the decision, which should be presented to the user. A possible strategy could be to select the most specific one. However, in the case of *What are the components of a car?*, the answer *molecules*[16] is probably not appropriate.

When facing with multiple correct answers at different levels of granularity, the decision about which to choose is fuzzy. This uncertainty is a source of problems in the evaluation of the systems' results, since even among human assessors there is often no agreement about what *is* the answer to a given question (Voorhees and Tice 1999). It also causes problems when building test collections for QA.

Lin and Katz (2006) focus on the granularity problem and discuss the characteristics of a correct answer. Granularity is seen as a critical topic because it makes the judgement dependent on the human assessor and on the question itself, specially if it belongs to the categories PERSON, LOCATION, or DATE. For instance, is a person's surname enough to answer a PERSON question, is a country enough to a LOCATION question, and is a year enough to a DATE question?

The work of Shanon (1979) on answers to *where*-questions allows to understand several criteria that can be met when dealing with answer granularity in QA. Based on experiments conducted with people from North America and Israel, the author discussed the granularity of answers as dependent on several factors. Namely, the reference points of questioner and answerer (for example, Americans give answers of a finer level of granularity than Israelis to the question *Where is the Empire State Building?*); the relative importance of an object toward another (for example, *Paris* is a more common answer than *France* to the question, *Where is the Eiffel Tower?*); and the importance of an object for the answerer (for example, Americans tend to locate Canada in relation to the United States of America).

Regarding the attempts to deal with the problem of granularity, Breck *et al.* (2000) suggest the definition of a set of strict directives to guide the evaluation of certain questions. The idea is to reduce uncertainty in evaluating answers and minimize the deregulation of the judgement.

On the one hand, the granularity of the cooperative answer depends on the question and on other factors that can be controlled with recourse to rules and guidelines, such as the questioner's position in space and, by analogy, in time. On

[16] This is certainly an anecdotal example, but it clearly demonstrates the impact of giving the user a correct answer to his/her question, however in a wrong level of granularity.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

the other hand, it should also consider the user who posed the question: his/her characteristics and expectations.

3.3 Useful answers

As opposed to what is expected to happen in the Dialogue Systems in which the interactions extend over multiple turns and where a broad variety of communication phenomena (speech acts) can be found, the typical interactions with QA systems are limited to isolated questions from the user side and answers from the system side – exceptions to this are the *interactive QA systems*, such as the systems developed in the context of the project Ritel (Rosset *et al.* 2006) and the HITIQA system (Small and Strazalkowski 2009). Therefore, the only information about the user profile must be obtained from the questions that he/she poses. Each question contains information by itself, but information can also be extrapolated from the current interaction context and, if kept, from the history of the user's previous interactions.

Systems can take into consideration clues about the user at the following three different levels:

- **Question clues**: Several clues can be obtained from the question, namely (and surely not limited to),
  - — the vocabulary that greatly differs depending on the user. For instance, on his/her age, academic background, and occupation;
  - — the specificity and world knowledge. If a user asks: *Who received the Prince of Asturias Award for Technical and Scientific Research for studies on the discovery of the first synthetic vaccination against malaria?*, probably he/she has deep knowledge of this specific field of science.
- **Clues in the current context**: Both questions and answers involved in the user–system interaction can provide some clues about his/her profile. For instance, consider a chain of questions about the Second World War. If the question *What is the real name of Dr. Death?* appears, probably the useful answer will be *Aribert Heim* or *Josef Mengele* (or both) and not any other from the set of possible answers.[17]
- **Clues in the history of interactions**: This can be seen as a general approach to the previous case, where the interactions with the users are kept, and the system is able to detect that he/she is interested in a certain topic.

User modelling and adaptation have been extensively studied in Artificial Intelligence as well as in the multidisciplinary field of Human–Computer Interaction. Also, it is not a new issue in the Information Retrieval, a field similar to QA, especially in web search engines that try to adapt the presentation of results to the user (see, for instance, the works of Liu, Yu and Meng 2004 and Teevand, Dumais and Horvitz 2005). However, first steps in this direction have been taken recently in QA.

[17] Note the multiple candidate answers for this question: `http://en.wikipedia.org/wiki/Dr._Death`

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

An example of user adaptation in QA is shown in DUARTE Digital (Mendes, Prada and Coheur 2009), an interactive QA system that considers the clues in the question and in the current context and uses both to adapt to the user. The DUARTE Digital answers questions about a piece of jewellery, and dynamically tries to assess its interlocutor characteristics, based on the used vocabulary. With a list of words that naive or expert users might employ, it interprets the user's expertise at every question and chooses the answer, previously marked with the corresponding difficulty level, from a knowledge base. Moreover, it also tries to acknowledge the user's goals at a contextual level, by measuring the proximity of the user's words in a sequence of questions to different sub-topics, in order to understand the orientation of the interaction. By doing so, it distinguishes from focused to stray interactions, and chooses the answer according to its details and informative level.

Quarteroni and Manandhar (2008) are also pioneers in user modelling. They added to the system YourQA the possibility to filter and re-rank the documents where answers are searched based on the proximity to the user's profile. The YourQA system deals with three parameters, explicitly provided to the system (it does not dynamically discover its interlocutor characteristics): the age range; the reading level; and the interests represented by a set of keywords extracted from the user's favorite documents and browsing history. The reading level was modelled as having a direct correspondence to each age range and is used to discard inappropriate documents. For that purpose, the reading level of all documents retrieved are estimated using the Smoothed Unigram Model learning approach. The third parameter is used to boost the relevance of the answers extracted from documents that match the user's preferences. Thus, every question submitted to the YourQA system can be answered by taking into account the user's model, but the user can still decide not to use this interaction mode. The YourQA shows a good reading level estimation with 94%, 85%, and 72% of the advanced, medium, and basic human-annotated passages correctly estimated, respectively; however, the authors point out the need of more studies to prove its usefulness in QA. Regarding the impact of interests in the YourQA, users acknowledged the system's adaptation to their profile, and considered the "user-adapted" answers more related to their interests and more useful than those found when the user's profile was not taken into account.

Besides the above-mentioned works, to our best knowledge, there is no other QA system that models the user and adapts the answer to him/her.

3.4 Discussion

Question-Answering systems are seen as intelligent machines that answer questions. Besides a couple of works in QA (such as the one by Quarteroni and Manandhar 2008), answers are given equal to everyone, regardless of the person who posed the question. If fact, this is sufficient in most cases. However, in our opinion, this represents a very simplistic view about the real users of QA systems.

In this section, we focus on what we believe to be the properties of a *better* answer, supporting ourselves in previous works on query answering. Our position is that, besides being correct, the answer should be useful – as it should take into account

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

the user's characteristics and address his/her specific needs – and non-misleading, as it should avoid misinterpretations.

The use of new information sources where systems search for the answer brought advantages and new challenges to QA. In fact, one of the issues raised in TREC had to do with the introduction of text from blogs, since these are often the source of answers that contradict answers found in other (more reliable) documents. In this context, we consider that credibility of the source where the answer was found is one of the conditions that has to hold for it to be considered as correct. Thus, our opinion is that the assessment of the credibility of the answer and its source is a necessary path to follow in order to deliver the user the correct answer to his/her question.

In other fields of knowledge, the importance of the user and his/her preferences is already fairly acknowledged. A practical example that gathers several advantages (including commercial) is the one of recommendation systems, where new items are picked and shown to the user based on his/her profile. In the particular case of QA systems, we consider that, in several cases, the answer has to be dependent on the person who posed the question. Even in the evaluation fora, the judgement is always dependent on the assessor's opinion about the correct answer to a question. Thus, we believe that attributing proper importance to the person behind the question is a necessary direction for future work in QA; this will have a positive impact on the users acceptance of systems.

4 Response generation

When it comes to responding to the user, and as we have previously mentioned, a myriad of systems choose to directly return the answer extracted from the information sources. The simplicity of this approach makes it especially attractive to the factoid QA. In fact, the factoid QA systems are expected to deliver short answers to questions, a trend in part promoted by current evaluation frameworks. However, even here, the snippet containing the answer has to be presented as a support (Dang *et al.* 2007; Forner *et al.* 2009).

Therefore, besides the answer that solves the question, which may or may not be the actual information chunk pinpointed from the information sources, the system's response often includes other textual elements. In this section, we refer to all the operations done on the system's answer as *response generation*. The generation of the response can imply the reformulation of the information chunk extracted from the information sources and/or the inclusion (removal) of textual information to (from) the answer.

There are several reasons that make the generation of a natural language response preferable to returning the extracted answer:

- It *humanizes* the system.
- It permits the usage of vocabulary and/or sentence constructions adapted to the user.
- It allows the introduction of additional information that the user did not explicitly request, but might be interested in.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

Giving a response to the user can be seen as a way to show cooperativeness. However, the generation of responses can only add value to the answering task if the final result is acceptable: for example, it makes no sense to return a syntactically flawed response when the direct answer did not have this problem. Benamara and Saint-Dizier (2004) explore this topic and identify a set of lexicalization strategies employed by humans, based on the analysis of a set of questions and their responses. The authors advocate that the natural language form of a response requires adequate and subtle lexicalizations. In certain circumstances, the meaning of a verb can be decisive for its usage over another verb: for instance, since *climb* indicates a faster growth than *go up*, the former is preferred for the generation of *The increase of gas prices climb to 20.3% in October 2005*, and the latter for *The increase of gas prices go up to 7.2% in September 2005* (Moriceau 2006).

The remainder of this section addresses the task of generating responses to questions in QA.

4.1 From answers to responses

The simplest generation approach to respond to the user's question is to directly retrieve the answer, that is, a chunk of information extracted from the information sources. Considering the definition of response generation as it was introduced, this can be seen as the application of the identity function to the answer. Typically, the factoid QA systems follow this approach.

Generation based on templates represents the next step of complexity in response generation. Templates realize the surface content of the response by directly mapping the non-linguistic content. Moriceau (2005) employs templates and makes the generation of responses to DATE questions dependent on the following variables:

- The characteristics of the event in question, namely, its recurrence (i.e., if it is unique or iterative) and regularity (i.e., if it is recurrent and happens at regular intervals).
- The type of answer (i.e., date or interval).
- A *certainty degree* calculated for the answer. This is a measure of the system's belief in the correctness of a certain answer and is expressed through the usage of adverbs and their intensities (e.g., possibly, most possibly, probably, most probably). With this, the user is given an indication of the confidence the system has in the answer that he/she has been given.

Another example of the usage of templates to generate responses in QA is the work by Benamara (2004a, 2004b), within the WEBCOOP system. The WEBCOOP system puts effort on showing cooperativeness to the user under two different perspectives: correcting misconceptions in his/her question (see Section 3.2) and intensional answering. An intensional answer represents a higher level of abstraction than the extensional one, and provides a description of the instances that satisfy the question by terms of the distinguishing properties they share, which is particularly important when the cardinality of the set of instances in the extensional answer is too high. The author describes an approach to create intensional answers from

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

the elements that compose the extensional answer to a question. First, it finds the elements to be generalized, using either

- the focus of the question: in *What are the means of transportation to go to Geneva airport?*, the focus *means of transportation* generalizes the elements *buses*, *trolleys*, and *trains*;
- a property of the focus of the question: in *What are the hotels at the border of the sea in Monaco?*, the property *hotel localization*, variable among all possible elements, is used; or
- a property in a given ontology of the focus of the question: in *Which hotels in Cannes have a swimming pool?*, the property *hotel category* serves as a generalizer.

Then, the set of intensional answers is augmented with the repeated grouping of the generalized elements with respect to the ontology. Finally, the best abstraction level is chosen through a *variable depth intensional calculus* that makes the decision depend on the ontological proximity of the elements in the intensional answer.

After the content has been determined, the WEBCOOP system builds natural language responses composed of two parts (Benamara and Saint-Dizier 2003). The first part indicates the misconceptions found in the user question with respect to the domain knowledge. Here, the authors indicate as main challenges the introduction of negations and the proper lexicalization, for what they prefer to keep negation to a minimum and use as many terms of the user question as possible. The second part shows the list of intensional answers to the user question or, in case of a misconception, presents the user with solutions to correct it. Here, the generation relies on a set of predetermined phrases and templates, instantiated with domain information. Moreover, the response is built with hyperlinks that allow the user to get further information on specific domain entities.

More sophisticated techniques to response generation in QA usually rely on *summarization*: the process of digesting information from texts, transforming them into short summaries that can be read and interpreted quicker. Summarization is often applied in the context of definitional questions,[18] such as *Who is X?* or *What is Y?*, which expect longer and explanatory answers, and for which the model employed in factoid questions, based on the retrieval of a short answer, does not apply.

Ye, Chua and Lu (2009) describe an approach to build summaries of different lengths using the texts of Wikipedia articles and the information contained in the infoboxes. The authors developed an extended document concept lattice model, able to select and group sentences representative of local topics. These sentences are then used as answers.

The DefScriber (Blair-Goldensohn, Mckeown and Schlaikjer 2003) system uses a hybrid approach to answer definitional, biographical, and topic-focused questions, borrowing techniques from summarization and generation. The authors introduce the concept of *definitional predicates* as the information types that are often useful

[18] Here we broaden the scope of this survey by referring strategies that aim at responding definitional questions.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

in definitions; for instance, the *Genus* predicate captures the category to which the term belongs. The DefScriber starts by applying a goal-driven strategy to capture those predicates from the text that are based on a machine-learning classifier and syntactic pattern matching; afterwards a data-driven strategy infers and organizes the themes that may arise from the data.

4.2 Answer support

Till this point, the focus of this survey was on how to respond to a question, either in the form of an information chunk pinpointed from the information sources, or in the form of a linguistically manipulated and generated response. However, state-of-the-art query/QC systems[19] often do more than giving the response to the user: They introduce a short summary of the document where the answer was found, highlighting the tokens of the user's query. Wu, Radev and Fan (2002) refer to these responses as *answer-focused summaries*, defining them as a specific type of document summaries that aim at providing answers to user's questions. Answer-focused summaries have to satisfy three criteria: (1) accuracy: they must have the answer to the question; (2) economy: they should be concise; and (3) support, they should substantiate the answer.

In QA, Lin *et al.* (2003) focused on the role of the surrounding context to make a 'good answer'. Lin and colleagues varied the length of the text surrounding the correct answer to several questions – *exact answer*, *answer-in-sentence*, *answer-in-paragraph*, and *answer-in-document*. The users showed preference for the *answer-in-paragraph*, which was considered 'good size of information chunk', while the exact answer was considered a too short chunk and the entire document too long, in line with the belief that the most natural response presentation style is focus-plus-context (Leung and Apperley 1994). The *answer-in-sentence* did not seem to bring any advantage over the *exact answer*, and posed some issues related with information taken out of context, specially due to the presence of anaphoric expressions.

Responses augmented with content can improve user's experience with the system in three different ways:

- By providing a justification why a certain answer was given.
- By allowing the user to validate the answer he/she was given.
- By offering extra information that can be of interest to the user.

Many QA systems built responses to the user's questions by adding contextual information to the answer. The added contextual information can be domain-specific, fetched from an ontology. For example, the AQUA system (Vargas-Vera and Motta 2004) responds to *Who works in AKT?* with additional contextual information about the entities under consideration: *AKT is a project at KMi and each person of the AKT team is a researcher at KMi.* However, the contextual information is typically retrieved from the sources where the answers were found.

[19] We refer to query/QA in order to cover search engines, such as Google or Yahoo!, that also accept a set of keywords as input.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

The QANDA system (Breck *et al.* 1999) scores each sentence according to the number of overlapping tokens with the question, and the best scored is added to the answer: The system returns *I.M. Pei: He designed the Hancock Building.* as response to the question *Who designed the Hancock Building in Boston?*. This strategy presupposes a phase of coreference resolution to assure that the entity in the answer and supporting sentence is the same.

The system of Bosma (2005) relies on the rhetorical structure theory (RST) (Mann and Thompson 1988), a theory that describes the organization of texts built around the notions of *nucleus* – a text span that better represents the writer's intention – and *satellite* – a text span whose purpose is to increase user's understanding – and their relations. Applied to QA, the nucleus is the answer and the satellite is the context. Bosma (2005) creates weighted graphs from the RST structures of the sentences in a text. With this, the distances between individual sentences are calculated and the closest sentences to the answer sentence are chosen as context.

In the clinical domain, Demner-Fushman and Lin (2006) present a hybrid approach to response generation that borrows techniques from information retrieval and summarization. First, the answer extraction phase picks the answers from MEDLINE citations retrieved by the PubMed[20] search engine. Afterwards, a semantic clustering phase groups the retrieved citations; finally, the extractive summarization phase creates summaries from each citation. These summaries are the support of each answer.

Other systems specifically address the problem of justifying the answers given to the user. An example is the MySentient Answers system (Curtis, Matthews and Baxter 2005) that uses the assertions in the knowledge base that lead to each answer. The responses are built with recourse to generation templates associated with each functor. These templates contain syntactic and semantic information and allow to grammatically manipulate and create the system's response.

4.3 Discussion

In this section we revisit several techniques for generating responses. Our position is that, in several situations, addressing the posed question with a generated response can be preferable to returning uniquely an answer extracted from the information sources. For example, in case the system is not totally confident on the correctness of the extracted answer, it is more appropriate to indicate to the user that there is a possibility that the answer might be wrong.

Currently, the success of a QA system is entirely dependent on the correctness of the returned answer: If the system returns a correct answer, we consider that it was successful; if it returns a wrong answer, we say that it failed its job.

We believe that a generated response can positively influence the acceptance of the system by real users, as they are highly permeable to the system's mistakes. On the one hand it provides a way for the user to confirm the given answer, which can be critical in some domains, such as the medical domain, and it is also a means

[20] Available at `http://www.ncbi.nlm.nih.gov/pubmed` Last accessed in November 2011.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

to demonstrate the system's cooperativeness. On the other hand it is also a way to discard responsibilities in case the answer is wrong. If the system returns a wrong answer because the information source from where it was extracted is mistaken, showing that paragraph to the user can minimize his/her negative reaction to the wrong answer while justifying why it was given.

Nevertheless, the response will only give credibility to the system if it is correctly generated, as it has to be flawless at the linguistic level.

5 Conclusions and future directions

Research in QA has been evolving in a way that systems try to deal with more and more complex questions. From answering *why* questions (Verberne 2009) to opinion questions (Balahur *et al.* 2010), for instance. In this survey we went back to the problem of answering factoid questions, as we believe that finding correct answers to these types of questions is still an open research problem.

To address the user's question, several open-domain QA systems follow a strategy that is based on direct extraction of answers from document collections or the web, supported by the redundancy of the used information sources. However, as we have seen throughout this paper, it has some flaws. For instance, instead of working jointly for the selection of the final answer, nonequivalent candidate answers are usually considered separately, even if they are semantically related – the point of view that assumes that the candidate answers are competitors, rather than allies (Dalmas and Webber 2007). Moreover, important procedures to ascertain that the user understands the answer are often neglected; for example, it is not taken into consideration whether the answer is ambiguous, or even if it contains inadequate vocabulary.

This survey covered current efforts regarding the answering task in QA. We started from the answer extraction and selection approaches based on answer redundancy and investigated current works on three different research lines.

First, we addressed relations existing between candidate answers for the purpose of computing the final answer. In this context, we presented a typology of four general relations (Webber *et al.* 2002) and described several techniques for their detection. We showed the work of Dalmas and Webber (2007) as an example of a system that considers candidate answers as allies instead of competitors, and relate them by equivalence and inclusion. Afterwards, we recovered the notion of cooperative answer (Kaplan 1979) as being a correct, useful, and non-misleading answer (Gaasterland *et al.* 1992). Besides surveying what makes an answer correct, we discussed a different model for QA systems that give the user an individualized (i.e., useful) answer with an unique interpretation (i.e., non-misleading). As an example, we presented the YourQA system (Quarteroni and Manandhar 2008), a system that adapts its answer to the current user's profile. Finally, we considered the generation of answers as a complementary approach to using an information chunk directly extracted from the corpus and described from template- to summarization-based generation strategies. Moreover, we presented systems that build answer with extra content that can be of user's interest. We gave the example of the work by

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

Moriceau (2005) that uses generation to provide user a hint about the system's confidence on the answer.

Question-Answering is a very appealing NLP task that has gained the interest of the research community. Much work was done in QA, but there is definitely much more to do. Recently, a group of researchers debated the QA technology, where it is now, and where it is heading. This exercise resulted in a white paper (Ferrucci *et al.* 2009) that promotes the openness of the QA technology to a larger community, including business partners and the overall society. The authors of the paper point to several major targets regarding the future of the research in QA; for instance, the establishment of practices that allow to share and reuse techniques and implementations – open-sourcing or API specification – and the setting of evaluation criteria against which the performance can be measured in a transparent and reproducible way, which includes the following dimensions: accuracy in the top N retrieved answers (such as in the current evaluation frameworks), the degree of confidence of the system in the returned answer, or the required level of user interaction/usability.

After exploring the literature and writing this survey, it is our opinion that research in the specific task of answering in QA systems should evolve in three different directions. In the following we present these directions and make a bridge between these and performance dimensions introduced in the white paper by Ferrucci *et al.* (2009):

Answer credibility and justification. The web is a resource increasingly used in QA. Systems have to overcome the existence of fake or flawed information, which is typically done with recourse to redundancy. Nevertheless, the incorporation of more robust mechanisms to guarantee the credibility of the answer are required, as well as the justification of how and why that answer was given. This direction is related with the dimensions of accuracy and degree of confidence (of the system in the answer) of the white paper.

User adaptation. QA systems are built for humans. However, QA systems did not achieve the impact on users as information retrieval systems did, with search engines, such as Google or Yahoo!, already becoming part of people's lives. We believe that one way to make QA an appealing alternative for users has to do with personalisation. In this sense, the path is still long with a few open research questions: Which characteristics of users should be modelled? How to dynamically infer a user model from posed questions? How to effectively present results according to the user? Should the personalisation be limited, or will it be acceptable that each user receives a different answer? This direction is related with the usability dimension of the white paper, as we consider the QA system as a tool to be used by humans whose information needs have to be satisfied and whose individual characteristics should be considered.

Systems evaluation. Theoretically, cooperativeness arises as a desirable model for QA. In this context, the definition of the evaluation criteria of QA systems is necessary, allowing both to measure the performance of a system, as well as further comparisons between systems. This direction is related with the

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

evaluation of cooperativeness in QA, thus falling in the usability dimension of the white paper.

Finally, it is our opinion that the successful strategies implemented in other fields of knowledge can also be used in QA, as throughout the paper we have mentioned works and techniques in different areas of NLP that can be brought to QA.

Acknowledgments

This work was partially supported by Fundação para a Ciência e a Tecnologia (INESC-ID multiannual funding) through the PIDDAC Program funds.

Ana Cristina Mendes is supported by a PhD fellowship from Fundação para a Ciência e a Tecnologia (SFRH/BD/43487/2008).

The authors thank to referees for their valuable comments and to Ângela Costa and Bruno Martins for their reviews and comments.

References

Androutsopoulos, I., Ritchie, G. D., and Thanisch, P. 1995. Natural language interfaces to databases—an introduction. *Natural Language Engineering* **1**: 29–81.

Balahur, A., Boldrini, E., Montoyo, A., and Martínez-Barco, P. 2010. Going beyond traditional QA systems: challenges and keys in opinion question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics COLING '10: Posters*, Beijing, China, pp. 27–35. Stroudsburg, PA: ACL.

Benamara, F. 2004a. Generating intensional answers in intelligent question answering systems. In A. Belz *et al.* (eds.), *International Conference on Natural Language Generation (INLG)*, pp. 11–20. Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag.

Benamara, F. 2004b. Cooperative question answering in restricted domains: the WEBCOOP experiment. In D. Mollá and J. L. Vicedo (eds.), *ACL 2004: Question Answering in Restricted Domains*, pp. 31–8. Stroudsburg, PA: ACL.

Benamara, F., and Saint-Dizier, P. 2003. Dynamic generation of cooperative natural language responses. In *Proceedings of the 9th European Workshop on Natural Language Generation (ENLG-2003)*, Budapest, Hungary, pp. 56–67. Stroudsburg, PA: ACL.

Benamara, F., and Saint-Dizier, P. 2004. Lexicalisation strategies in cooperative question answering systems. In B. Gambäck and K. Jokinen (eds.), *Proceedings of the 20th International Conference on Computational Linguistics*, (*COLING '04*) pp. 1179–85. Stroudsburg, PA: ACL.

Blair-Goldensohn, S., Mckeown, K. R., and Schlaikjer, A. H. 2003. A hybrid approach for QA track definitional questions. In *Proceedings of the 12th Annual Text Retrieval Conference* (*TREC 2003*), Gaithersburg, MD, USA, pp. 185–92.

Bos, J., Curran, J. R., and Guzzetti, E. 2007. The pronto QA system at TREC-2007: harvesting hyponyms, using nominalisation patterns, and computing answer cardinality. In E. M. Voorhees and L. P. Buckland (eds.), *Proceedings of the 16th Text RETrieval Conference, TREC 2007*, Gaithersburg, MD, USA, pp. 726–32.

Bosma, W. E. 2005. Extending answers using discourse structure. In H. Saggion and J. L. Minel (eds.), *RANLP Workshop on Crossing Barriers in Text Summarization Research*, Borovets, Bulgaria, pp. 2–9.

Breck, E. J., Burger, J. D., Ferro, L., Hirschman, L., House, D., Light, M., and Mani, I. 2000. How to evaluate your question answering system every day and still get real work done. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece, pp. 1495–500.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

Breck, E. J., Burger, J., Ferro, L., House, D., Light, M., and Mani, I. 1999. A sys called qanda. In E. M. Voorhees and D. K. Harman (eds.), *Proceedings of the 8th Text REtrieval Conference (TREC 8)*, Gaithersburg, MD, pp. 499–506.

Brill, E., Dumais, S., and Banko, M. 2002. An analysis of the AskMSR question-answering system. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing – Volume 10*, pp. 257–64. Stroudsburg, PA: ACL.

Brill, E., Lin, J., Banko, M., Dumais, S., and Ng, A. 2001. Data-intensive question answering. In *Proceedings of the 10th Text REtrieval Conference (TREC 2001)*, ithersburg, MD, pp. 393–400.

Buchholz, S., and Daelemans, W. 2001. Complex answers: a case study using a WWW question answering system. *Natural Language Engineering* **7**: 301–23.

Cardie, C., Ng, V., Pierce, D., and Buckley, C. 2000. Examining the role of statistical and linguistic knowledge sources in a general-knowledge question-answering system. In *Proceedings of the 6th Conference on Applied Natural Language Processing*, pp. 180–87. Stroudsburg, PA: ACL.

Chklovski, T., and Pantel, P. 2004. VerbOcean: mining the web] for fine-grained semantic verb relations. In D. Lin and D. Wu (eds.), *Proceedings of the EMNLP 2004*, pp. 33–40. Stroudsburg, PA: ACL.

Clarke, C. L. A., Cormack, G. V., and Lynam, T. R. 2001. Exploiting redundancy in question answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM.

Curtis, J., Matthews, G., and Baxter, D. 2005. On the effective use of cyc in a question answering system. In *Proceedings of the IJCAI Workshop on Knowledge and Reasoning for Answering Questions*, Edinburgh, Scotland, pp. 61–70.

Dalmas, T., and Webber, B. 2005. Using information fusion for open domain question answering. *Proceedings of the KRAQ Workshop, IJCAI*, Edinburgh, Scotland.

Dalmas, T., and Webber, B. 2007. Answer comparison in automated question answering. *Journal of Applied Logic* **5**: 104–20.

Dang, H. T., Kelly, D., and Lin, J. J. 2007. Overview of the TREC 2007 question answering track. In E. M. Voorhees and L. P. Buckland (eds.), *Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007*, Gaithersburg, MD, pp. 137–45. Gaithersburg, MD: NIST.

De Boni, M., and Manandhar, S. 2003. An analysis of clarification dialogue for question answering. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology – Volume 1*, pp. 48–55. Stroudsburg, PA: ACL.

De Marneffe, M., Rafferty, A. N., and Manning, C. D. 2008. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pp. 1039–47. Stroudsburg, PA: ACL.

Demner-Fushman, D., and Lin, J. 2006. Answer extraction, semantic clustering, and extractive summarization for clinical question answering In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pp. 841–8. Stroudsburg, PA: ACL.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Ferrucci, D., Nyberg, E., Allan, J., Barker, K., Brown, E., Chu-Carroll, J., Ciccolo, A., Duboue, P., Fan, J., Gondek, D., Hovy, E., Katz, B., Lally, A., McCord, M., Morarescu, P., Murdock, B., Porter, B., Prager, J., Welty, C., and Zadrozny, W. 2009. Towards the open advancement of question answering systems. IBM Technical Report, RC 24789, IBM Research Division, New York.

Forner, P., Peñas, A., Agirre, E., Alegria, I., Forăscu, C., Moreau, N., Osenova, P., Prokopidis, P., Rocha, P., Sacaleanu, B., Sutcliffe, R., and Sang, E. T. K. 2009. Overview of the Clef 2008 multilingual question answering track. In *Proceedings of the 9th Cross-language Evaluation Forum Conference on Evaluating Systems for Multilingual and Multimodal Information Access*, pp. 262–95. Berlin: Springer-Verlag.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

Gaasterland, T., Godfrey, P., and Minker, J. 1992. An overview of cooperative answering. *Journal of Intelligent Information Systems* **1**: 123–57.

Grice, H. 1975. *Logic and Conversation. Syntax and Semantics*, vol. 3, Speech Acts. Maryland Heights, MO: Academic Press.

Harabagiu, S., and Maiorano, S. 1999. Finding answers in large collections of texts: paragraph indexing + abductive inference. In *Proceedings of the AAAI Fall Symposium on Question Answering Systems*, Toulouse, France, pp. 63–71.

Harabagiu, S., Hickl, A., and Lacatusu, F. 2006. Negation, contrast and contradiction in text processing. In *AAAI'06: Proceedings of the 21st National Conference on Artificial Intelligence*, pp. 755–62. Palo Alto, CA: AAAI Press.

Harabagiu, S., Moldovan, D., Paşca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girju, R., Rus, V., Morerescu, P. 2000. FALCON – boosting knowledge for answer engines. *Proceedings of Text REtrieval Conference (TREC-9)*, Gaithersburg, MD.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, pp. 539–45. Stroudsburg, PA: ACL.

Hermjakob, U. 2001. Parsing and question classification for question answering. In *Proceedings of the Workshop on Open-Domain Question Answering – Volume 12*, pp. 1–6. Stroudsburg, PA: ACL.

Huang, Z., Thint, M., and Qin, Z. 2008. Question classification using head words and their hypernyms. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 927–36. Stroudsburg, PA: ACL.

Hung, J. C., Wang, C., Yang, C., Chiu, M., and Yee, G. 2005. Applying word sense disambiguation to question answering system for e-learning. In *AINA '05: Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pp. 157–62. Washington, DC: IEEE Computer Society.

Iftene, A., and Balahur, A. 2008. Answer validation on English and Romanian languages. In C. Peters *et al.* (eds.), *Evaluating Systems for Multilingual and Multimodal Information Access*, pp. 448–51. Lecture Notes in Computer Science. Berlin: Springer-Verlag.

Ittycheriah, A., Franz, M., and Roukos, S. 2001. IBM's statistical question answering system – TREC-10. *Proceedings of the 10th Text REtrieval Conference (TREC-10)*, Gaithersburg, MD.

Jurafsky, D., and Martin, J. H. 2008. *Speech and Language Processing*, 2nd ed. (Prentice Hall Series in Artificial Intelligence). New Jersey: Prentice Hall.

Kamp, H., and Reyle, U. 1993. *From Discourse to Logic: Introduction to Model-Theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Netherlands: Kluwer.

Kaplan, S. 1979. *Cooperative Responses from a Portable Natural Language Data Base Query System*. PhD thesis, University of Pennsylvania.

Kaplan, S. 1981. Appropriate responses to inappropriate questions. In *Elements of Discourse Understanding*, pp. 127–144. Cambridge, UK: Cambridge University Press.

Kehler, A. 2002. *Coherence, Reference and the Theory of Grammar*. Stanford, CA: CSLI.

Ko, J., Si. L., and Nyberg, E. 2007. A probabilistic framework for answer selection in question answering. In C. L. Sidner *et al.* (eds.), *Proceedings of Human Language Technology Conference of the NAACL*, pp. 524–31. Stroudsburg, PA: ACL.

Kozareva, Z., Vazquez, S., and Montoyo, A. 2006. Adaptation of a machine-learning textual entailment system to a multilingual answer validation exercise. In *Working Notes of CLEF–ECDL 2006, AVE Workshop,* LNCS 3237. Berlin: Springer-Verlag.

Kwok, C., Etzioni, O., and Weld, D. S. 2001. Scaling question answering to the web. *ACM Transactions on Information Systems* **19**: 242–62.

Leung, Y. K., and Apperley, M. D. 1994. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction* **1**: 126–60.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

Lin, J. 2007. An exploration of the principles underlying redundancy-based factoid question answering. *ACM Transactions on Information Systems* **25**: 1–55.

Lin, J., and Katz, B. 2006. Building a reusable test collection for question answering. *Journal of the American Society for Information Science and Technology* **57**: 851–61.

Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., and Karger, D. R. 2003. What makes a good answer? The role of context in question answering. In M. Rauterberg *et al.* (eds.) *Human-Computer Interaction (INTERACT 2003)*, Zurich, Switzerland, pp. 25–32.

Liu, F., Yu, C., and Meng, W. 2004. Personalized web search for improving retrieval effectiveness. *IEEE Transactions on Knowledge and Data Engineering* **16**: 28–40.

Lin, D., Zhao, S., Qin, L., and Zhou, M. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, pp. 1–7. Stroudsburg, PA: ACL.

Magnini, B., Negri, M., Prevete, R., and Tanev, H. 2002. Is it the right answer?: exploiting web redundancy for answer validation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 425–32. Stroudsburg, PA: ACL.

Magnini, B., Romagnoli, S., Vallin, A., Herrera, J., Peñas, A., Peinado, V., Verdejo, F., De Rijke, M., and Vallin, R. 2003. The multiple language question answering track at CLEF 2003. In *CLEF 2003. CLEF 2003 Workshop*.Berlin: Springer-Verlag.

Magnini, B., Vallin, R., Ayache, C., Erbach, G., Peñas, A., De Rijke, M., Rocha, P., Kiril, S., and Sutcliffe, R. 2005. Overview of the CLEF 2004 multilingual question answering track. In *Results of the CLEF 2004 Cross-Language System Evaluation Campaign*, pp. 371–91. Berlin: Springer-Verlag.

Mann, W. C., and Thompson, S. A. 1988. Rhetorical structure theory: toward a functional theory of text organization. *Text* **8**: 243–81.

Mendes, A. C., and Coheur, L. 2011. An approach to answer selection in question-answering based on semantic relations. In T. Walsh (ed.), *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1852–1857.

Mendes, A. C., Prada, R., and Coheur, L. 2009. Adapting a virtual agent to users' vocabulary and needs. In Rutkay *et al.* (eds.), *IVA '09: Proceedings of the 9th International Conference on Intelligent Virtual Agents*, pp. 529–30. Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag.

Moldovan, D., Clark, C., Harabagiu, S., and Maiorano, S. 2003. COGEX: a logic prover for question answering. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology – Volume 1*, pp. 87–93. Stroudsburg, PA: ACL.

Moldovan, D., Harabagiu, S., Paşca, M., Mihalcea, R., Girju, R., Goodrum, R., and Rus, V. 2001. The structure and performance of an open-domain question answering system. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 563–70. Stroudsburg, PA: ACL.

Moldovan, D., Paşca, M., Harabagiu, S., and Surdeanu, M. 2003. Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems* **21**: 133–54.

Mollá, D., and Gardiner, M. 2004. Answerfinder: question answering by combining lexical, syntactic and semantic information. In *Proceedings of the Australasian Language Technology Workshop 2004*, NSW, Australia, pp. 9–16.

Moriceau, V. 2005. Answer generation with temporal data integration. In G. Wilcock *et al.* (eds.), *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05)*, University of Aberdeen, Aberdeen, UK, pp. 197–202.

Moriceau, V. 2006. Numerical data integration for cooperative question-answering. In *Proceedings of the Workshop KRAQ'06 on Knowledge and Reasoning for Language Processing*, pp. 43–50. Stroudsburg, PA: ACL.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

Moriceau, V. 2007) *Intégration de Données dans un Système Question-Réponse sur le Web*. PhD thesis, Université Paul Sabatier, Toulouse, France.

Nii, Y., Kawata, K., Yoshida, T., Sakai, H., and Masuyama, S. 2004. Question answering system QUARK. *Proceedings of the NTCIR-4*, Tokyo, Japan.

Nyberg, E., Mitamura, T., Callan, J. P., Carbonell, J. G., Frederking, R., Collins-Thompson, K., Hiyakumoto, L., Huang, Y., Huttenhower, C., Judy, S., Ko, J., Kupsc, A., Lita, L. V., Pedro, V., Svoboda, D., and Van Durme, B. 2003. The JAVELIN question-answering system at TREC 2003: a multi-strategy approach with dynamic planning. In *Proceedings of the 2003 Text REtrieval Conference (TREC 2003)*, Gaithersburg, MD.

Nyberg, E., Mitamura, T., Carbonell, J. G., Callan, J. P., Collins-Thompson, K., Czuba, K., Duggan, M., Hiyakumoto, L., Hu, N., Huang, Y., Ko, J., Lita, L. V., Murtagh, S., Pedro, V., and Svoboda, D. 2002. The JAVELIN question-answering system at TREC 2002. In *Proceedings of the TREC'02*, Gaithersburg, MD.

Pantel, P., and Ravichandran, D. 2004. Automatically labeling semantic classes. In S. Dumais, D. Marcu, and S. Roukos (eds.) *Proceedings of the North American Association for Computational Linguistics and the Human Language Technologies Conferences (NAACL/HLT2004)*, pp. 321–28. Stroudsburg, PA: ACL.

Paşca, M., and Harabagiu, S. 2001. The informative role of WordNet in open-domain question answering. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-01), Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*. Stroudsburg, PA: ACL.

Quarteroni, S., and Manandhar, S. 2007. User modelling for personalized question answering. In R. Basili and M. Pazienza (eds.), *AI*IA '07: Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence on AI*IA 2007*, pp. 386–97. Lecture Notes in Computer Science. Berlin: Springer-Verlag.

Quarteroni, S., and Manandhar, S. 2008. Designing an interactive open-domain question answering system. *Journal of Natural Language Engineering* **15**: 73–95.

Ritter, A., Soderland, S., and Etzioni, O. 2009. What is this, anyway: automatic hypernym discovery. In *Proceedings of AAAI-09 Spring Symposium on Learning by Reading and Learning to Read*, pp. 88–93. Palo Alto, CA: AAAI.

Rodrigo, Á., Peñas, A., and Verdejo, F. 2009. Overview of the answer validation exercise 2008. In A. Belz *et al.* (eds.), *Proceedings of the 9th Cross-Language Evaluation Forum Conference on Evaluating Systems for Multilingual and Multimodal Information Access*, pp. 296–313. Berlin: Springer-Verlag.

Rosset, S., Galibert, O., Illouz, G., and Max, A. 2006. Integrating Spoken Dialog and Question Answering: The Ritel Project. In *Proceedings of Interspeech 2006—ICSLP: 9th International Conference on Spoken Language Processing*, Pittsburgh, PA, pp. 1914–17.

Schlobach, S., Ahn, D., de Rijke, M., and Jijkoun, V. 2007. Data-driven type checking in open domain question answering. *Journal of Applied Logic* **5**: 121–43.

Shanon, B. 1979. Where questions. In *Proceedings of the 17th Annual Meeting on Association for Computational Linguistics*, pp. 73–5. Stroudsburg, PA: ACL.

Silva, J., Coheur, L., Mendes, A., and Wichert, A. 2011. From symbolic to sub-symbolic information in question classification. *Artificial Intelligence Review* **35**: 137–54.

Small, S., and Strazalkowski, T. 2009. HITIQA: high-quality intelligence through interactive question answering. *Natural Language Engineering* **15**: 31–54.

Teevan, J., Dumais, S. T., and Horvitz, E. 2005. Personalizing search via automated analysis of interests and activities. In *SIGIR '05: Proceedings of the 28th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 449–56. New York: ACM.

Téllez-Valero, A., Juárez-González, A., Montes-y-Gómez, M., and Villaseñor-Pineda, L. 2008. INAOE at QA@CLEF 2008: evaluating answer validation in Spanish question

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press

answering. In *Working Notes of CLEF–ECDL 2008, AVE Workshop,* Aarhus, Denmark. Berlin: Springer-Verlag.

Téllez-Valero, A., Montes-Gómez, M., Villaseñor-Pineda, L., and Peñas, A. 2010. Towards multi-stream question answering using answer validation. *Informatica. Special Issue on Computational Linguistics and its Applications* **34**: 45–54.

Vargas-Vera, M., and Motta, E. 2004. AQUA " ontology-based question answering system. In *MICAI 2004: Advances in Artificial Intelligence*, pp. 468–77. Berlin: Springer.

Verberne, S. 2009. *In Search of the Why: Developing a System for Answering Why-Questions.* PhD thesis, University of Nijmegen, Nijmegen.

Voorhees, E. M., and Tice, D. M. 1999. The TREC-8 question answering track evaluation. In *In Text Retrieval Conference TREC-8*, Gaithersburg, MD, pp. 83–105.

Webber, B., Gardent, C., and Bos, J. 2002. Position statement: inference in question answering. In *Proceedings of the LREC Workshop on Question Answering: Strategy and Resources*, Canary Islands, Spain, pp. 19–25.

Wu, H., Radev, D. R., and Fan, W. 2002. Towards answer focused summarization. *Proceedings of the 1st International Conference on Information Technology and Applications.*

Ye, S., Chua, T., and Lu, J. 2009. Summarizing definition from Wikipedia. In *ACL-IJCNLP '09: Proceedings of the Joint Conf. of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pp. 199–207. Stroudsburg, PA: ACL.

https://doi.org/10.1017/S1351324911000350 Published online by Cambridge University Press