How to Make Right Decisions Based on Corrupt Information and Poor Counselors

Tuning an Open-Source Question Answering System

Michael Muck[1,2] and David Suendermann-Oeft[1]

[1] DHBW Stuttgart, Stuttgart, Germany**
[2] Tesat-Spacecom GmbH & Co. KG, Backnang, Germany

Abstract. This paper describes our efforts to tune the open-source question answering system OpenEphyra and to adapt it to interface with multiple available search engines (Bing, Google, and Ixquick). We also evaluate the effect of using outdated test data to measure performance of question answering systems. As a practical use case, we implemented OpenEphyra as an integral component of the open-source spoken dialog system Halef.

1 Introduction

Over the last years, academic and industrial interest in automatic question answering technology has substantially grown [1]. In addition to commercial implementations question answering engines such as IBM's Watson DeepQA [2] or Wolfram Alpha [3], there is a number of academic engines such as QA-SYS [4], OSQA [5], and OpenEphyra [6]. Being an open-source software, the latter turned out to be particularly suitable for a number of applications the DHBW Spoken Dialog Systems Research Center is working on, such as the free spoken dialog system Halef [7].

To make sure OpenEphyra meets the quality standards required for an integration into Halef, we undertook a thorough quantitative assessment of its performance. In order to do so, we used a standard test set for question answering (the NIST TREC-11 corpus) which raised a number of issues with respect to the consistency of test sets in the question answering domain, discussed further in Section 2.

Even though OpenEphyra is a free software, its original implementation was based on Bing API [8], a commercial service provided by Microsoft. To become independent of a certain provider, we implemented APIs to interface with a number of regular web search engines (Google, Bing, Ixquick). One of our major interests was to understand how performance depends on the specific search engine used and whether system combination would result in performance gain. We also analyzed the impact of multiple parameters, for instance associated with the number of queries per question or the number of search results taken into

** David Suendermann-Oeft is now with ETS, San Francisco, USA

account to compile the final system response. These activities are described in Section 3.

Finally, we describe our efforts to embed OpenEphyra as a webservice used as component of the dialog manager in the spoken dialog system Halef, a telephony-based distributed industry-standard-compliant spoken dialog system in Section 4.

2 Reflections on the Test Set

2.1 Measuring performance of questions answering systems

A common method to measure the performance of a question answering system is to use a test corpus of a certain number of questions each of which is associated with a set of possible correct (canonical) answers[3]. The system will produce a first-best response for all the involved questions which are then compared to the set of canonical answers. If, for a given question, the response is among the canonical answers, this event is considered a match. Ultimately, the total number of matches is divided by the total number of questions resulting in the questions answering accuracy [9].

For the current study, we used the NIST TREC-11 corpus (1394-1893) [10], a standard corpus provided by the National Institute of Standardization and Technology which had been used by IBM's statistical question answering system [11].

Corpus statistics are given in Table 1

Table 1. Statistics of the NIST TREC-11 corpus (1394-1893)

#questions	500
avg #answers	1.06
#questions w/o answer	56

2.2 Missing the correct answer

It is perceivable that, at times, the set of canonical answers does not contain the exact wording of system response which turns out to be correct. In such a case, it will be erroneously counted as an error negatively affecting accuracy. Even worse, in contrast to most other classification problems where the ground thruth is valid once and forever, the answer to certain questions is time-dependent (for example when asking for the location of the next World Cup). A detailed analysis of possible reasons for this lack is given in the following.

[3] There can be multiple ways to define this set including a simple list, regular expressions, or context-free grammars

Time dependence Answers may be obsolete:

Example 1. "Who is the governor of Colorado?"

- John Hickenlooper
- Bill Ritter

Missing answers There might be a multitude of terms referring to the same phenomenon:

Example 2. "What is the fear of lightning called?"

- astraphobia
- astrapophobia
- brontophobia
- keraunophobia
- tonitrophobia

Scientific ambiguity Different studies may provide different results:

Example 3. "How fast does a cheetah run?"

- 70 mph (*discovery.com*)
- 75 mph (*wikipedia.com*)

Degree of detail Some questions do not clearly specify how detailed the answer should be. In human interaction, this issue is resolved by means of a disambiguation dialog, or avoided by taking the interlocutor's context awareness into account:

Example 4. "How did Eva Peron die?"

- death
- disease
- cervical cancer

Example 5. "Where are the British Crown jewels kept?"

- Great Britain
- London
- Tower of London

Different units. There may be differences in physical units (e.g. metrical vs. US customary systems)

Example 6. "How high is Mount Kinabalu?"

- 4095 meter
- 4.095 kilometer
- 13,435 feet

Numerical precision. Not only when asking for irrational numbers such as pi, the precision of numerical values needs to be accounted for:

Example 7. "What is the degree of tilt of Earth?"

- 23.439 degrees
- 23.4 degrees
- 24 degrees

Partial answers. Some answers consist out of more than just one word, but for recognizing if this answer is correct there is no need for specifying all parts. E.g., when it comes to person names, it is often acceptable to return only certain parts:

Example 8. "Who was the first woman to run for president?"

- Victoria Claflin Woodhull
- Victoria Woodhull
- Victoria
- Woodhull

2.3 Effect on the TREC-11 corpus

After taking a close look at the TREC-11 test set, we had to rectify the set of canonical answers of as many as 120 of the 500 questions, that is about one quarter. To provide an example figure: The described corpus rectification improved one OpenEphyra test configuration (Bing1q) from 37.6% (188/500) to 55.8% (269/481) accuracy.

Obviously, the results of the present study cannot be directly compared to publications of prior publications using TREC-11. However, this would not have been possible anyway given the aforementioned phenomenon of obsolete answers. As a consequence, generally, evaluation of accuracy of question answering systems is a time-dependent undertaking, in a way comparable to measuring performance of, say, soccer teams.

3 Evaluation

After providing a short overview about the architecture of OpenEphyra, details of our enhancements and tuning results are presented.

3.1 A brief overview of OpenEphyra's architecture

Given a candidate question, OpenEphyra parses the question structure and transforms it into what resembles skeletons of possible statements containing the sought-for answers. These skeletons are referred to as *queries*. For instance:

question: When was Albert Einstein born?
answer: Albert Einstein was born in X
Albert Einstein was born on X.

At the same time, the answer type is identified (in this above case, it is a *date*). Then, OpenEphyra searches for documents matching the queries using a search API (OpenEphyra 2008-03-23 used the Bing API). After extracting matching strings from the documents and isolating the respective response candidates (X), a ranking list is established based on the count of identical candidates, metrics provided by the search API, and other factors which, combined in a certain fashion, constitute a confidence score. Finally, the first best answer is returned along with its confidence score [12]. Figure 1 provides a schema of the described process. The initial implementation of OpenEphyra was based on the Bing API

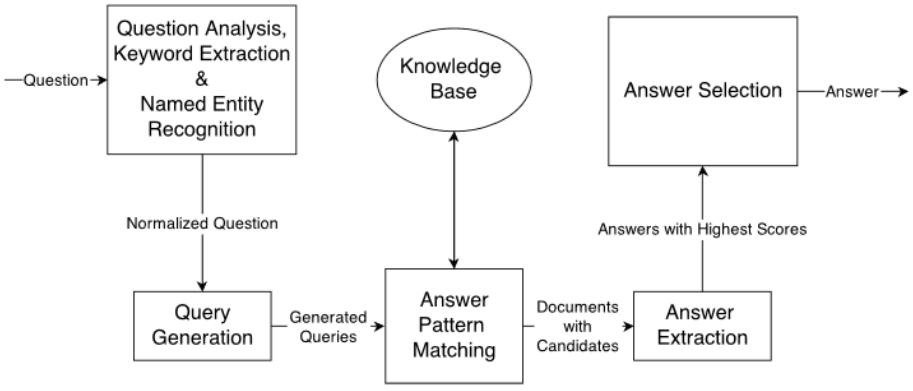

Fig. 1. OpenEphyra's principle architecture

requiring a subscription with a limit of 5000 gratis queries per month. This was the main motivation why we implemented interfaces for communication with regular search engines (Google, Bing, and Ixquick). Drawback of these APIs is that, in contract to the official Bing API, they do not have access to the search engines' confidence scores but only to the order of search results. Whereas Google and Bing, limit the number of requests per time unit to a certain degree to prevent abuse by webcrawlers, Ixquick turned out to be rather tolerant in this respect which is why it became our tool of choice.

3.2 Search engines, number of queries, and number of documents

First and foremost, we sought to find out which impact the use of web search engines has when compared to the native Bing API. When testing the latter against the TREC-11 corpus, we achieved a benchmark of 57.2% accuracy. It should be noted that OpenEphyra's default settings do not limit the number of queries per question. That is, depending on the question type, a large number of queries might be generated, all of which will be executed. On average, OpenEphyra produces 7.7 queries per question on the TREC-11 corpus. Furthermore, by default, 50 documents are retrieved per query. Table 2 shows accuracy results of multiple combinations of search engines, maximum number of queries, and number of retrieved documents. As aforementioned, due to the limited number of total queries provided by the Google and Bing engines, we were unable to increase the number of queries per question to more than two and the number of documents to more than ten.

Table 2. Experimental results

ID	**engine**	**#queries**	**#documents**	**#correct**	**accuracy/%**
BingAllq	Bing	∞	50	275	57.2
Ixquick200	Ixquick	∞	200	270	56.1
Bing1q	Bing	1	50	269	55.9
Ixquick100	Ixquick	∞	100	267	55.5
Bing3q	Bing	3	50	265	55.1
Bing2q	Bing	2	50	263	54.7
Ixquick50	Ixquick	∞	50	258	53.6
Ixquick20	Ixquick	∞	20	253	52.6
Google2q	Google	2	10	247	51.4
IxquickAllq	Ixquick	∞	10	243	50.5
Ixquick1q	Ixquick	1	10	235	48.9
Google1q	Google	1	10	233	48.4
Ixquick2q	Ixquick	2	10	225	46.8
BingW1q	BingW	1	10	202	42.0
BingW2q	BingW	2	10	202	42.0

Looking at the native Bing API as well as at Ixquick, it can be observed that increasing the number of queries per question from 1, 2, or 3 to unlimited (∞) has a positive impact on accuracy which, however, is not found to be statistically significant on the TREC-11 set (minimum p-value 0.25). Clearly more significant is the impact of the number of documents per question. As an example, Figure 2 shows this relationship for the Ixquick engine. The maximum number of queries per question was unlimited. The accuracy improvement from 10 to 200 considered documents per query was significant with a p-value of 0.08.

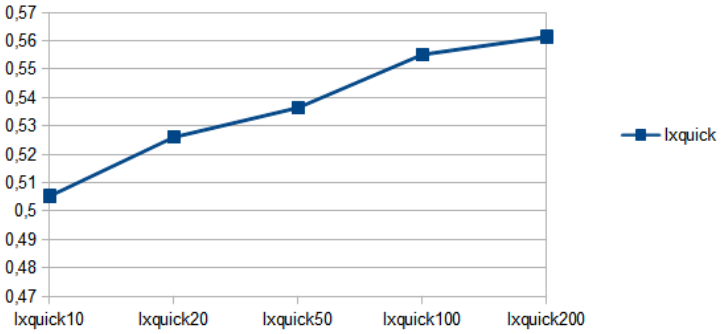

Fig. 2. Dependency of accuracy on the number of retrieved documents.

3.3 Answer type dependence

As described in Section 3.1, OpenEphyra generates queries depending on the detected answer type. Hence, we were interested to see how question answering performance depends on the answer type. Figure 3 shows the average performance of Open Ephyra over all test settings mentioned per each of the five main answer types.

- location,
- number,
- names (person/company/nickname/group),
- date,
- rest.

While location, names, and date perform pretty decently (60 to 70%), number and other questions perform at less than 40% accuracy. The reason for this are, among others, issues discussed in Section 2.2 (scientific ambiguity, different units, numerical precision).

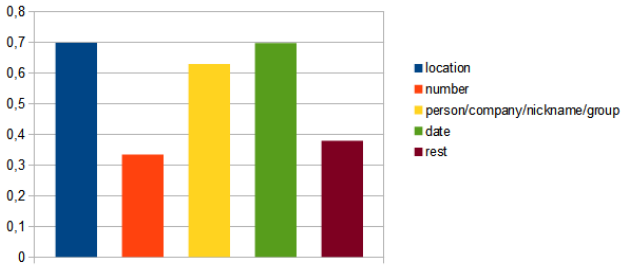

Fig. 3. Dependency of performance on answer types

Next, we wanted to see how question type dependence is influenced by which specific system (search engine, number of queries and documents per question) is used. Figure 4 provides an overview of the same systems discussed in Section 3.2.

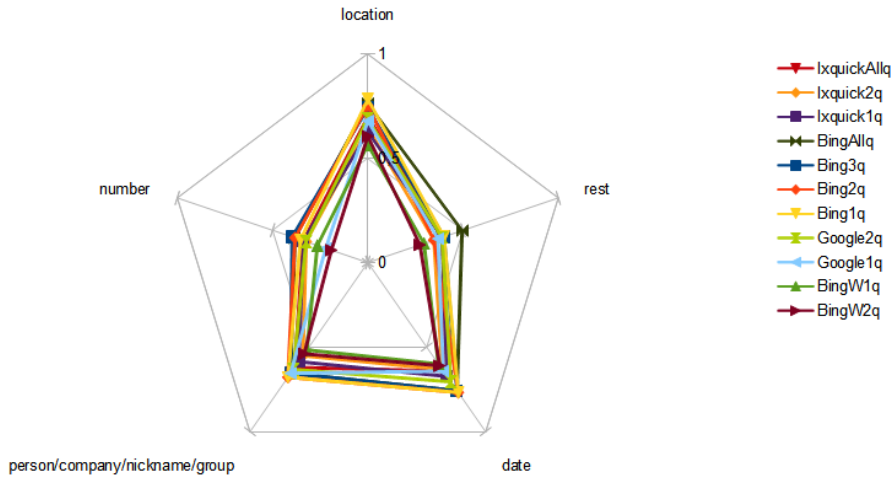

Fig. 4. netdiagram diversity

3.4 System Combination

Looking at the previous line of experiments, it is obvious that systems behave differently for different answer types. For example, the Google1q system performs decent on names, poor on numbers, and average on the other categories. In contrast, Ixquick2q performs poor on names and much better on numbers. This observation inspires the use of system combination to stimulate synergetic behavior.

When combining systems, we had to make sure that the n-best list of answers produced by the systems are identical. If an answer included in the n-best list of System A was missing in the n-best list of System B, it was included in the latter with a confidence of 0. Then, the individual n-best lists were merged multiplying the individual confidence scores with system-dependent weights and summing the results up across systems. This resulted in final n-best list the highest confidence result was considered winner. The system weights are greater or equal to zero and need to sum up to one. An example for tuning the weight of the system Ixquick20q when combined with Ixquick200q is shown in Figure 5. The peak is found at a weight of 0.3 where the system performs at 57.4% compared to the individual systems (52.6% and 56.1%). Minimum p-value is 0.14, that is, the effect is moderately statistically significant.

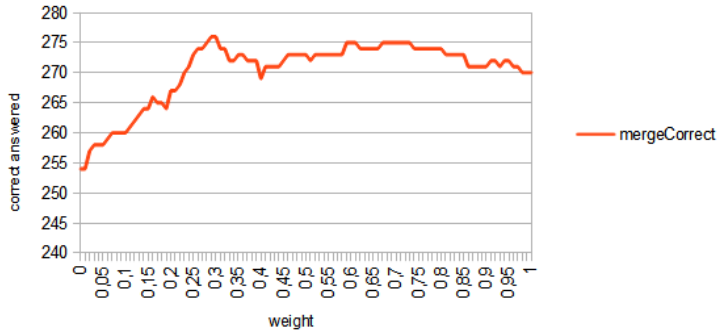

Fig. 5. system combination w. Ixquick20 & Ixquick200

4 Implementation in Halef

To demonstrate OpenEphyra's capabilities, we integrated it into the spoken dialog system Halef [7], an open-source industry-standard-compliant, distributed system. The system can be called using the free-of-charge number

```
+1-206-203-5276 Ext 2000
```

As the current system is limited to rule-based (JSGF) grammars, the set of possible questions that can be asked is limited to the ones encoded in the grammar, e.g.

Who discovered gravity?
When was Albert Einstein born?
Who invented the automobile?

5 Conclusion and Future Work

This paper features a number of contributions:

- We discussed the issue of outdated test sets in question answering, analyzed reasons, and possible ways of remedy.
- We presented the results of a number of tuning experiments with the open-source question answering system OpenEphyra.
- We desribed how we changed OpenEphyra's interface to external knowledge bases from a commercial search API to the direct connection to the web search engines Google, Bing, and Ixquick.
- We showed how by extensive parameter tuning and system combination, the new web search interface can perform en par with the original implementation based on a commercial search API.

In the future, we aim at addressing underperforming answer types (in particular numbers and, to some extent, names) and breaking the rest group down into multiple sub-groups each of which can be tackled independently [13].

6 Acknowledgements

We would like to thank all the members of Spoken Dialog Systems Research Center at DHBW Stuttgart. Furthermore, we wholeheartedly appreciate the continuous support of Patrick Proba and Monika Goldstein.

References

1. Strzalkowski, T., Harabagiu, S.: Advances in Open Domain Question Answering. Springer, New York, USA (2007)
2. Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, W., Nyberg, E., Prager, J., Schlaefer, N., Welty, C.: Building Watson: An Overview of the DeepQA Project. AI Magazine **31**(3) (2010)
3. Wolfram Alpha Champain, USA: Wolfram Alpha—Webservice API Reference. (2013) http://products.wolframalpha.com/docs/WolframAlpha-API-Reference.pdf.
4. Ng, J.P., Kan, M.Y.: Qanus: An open-source question-answering platform (2010)
5. Hattingh, F., Van Rooyen, C., Buitendag, A.: A living lab approach to gamification of an open source q&a for tertiary education. In: 2013 Conference. (2013)
6. N Schlaefer, E Nyberg, J.C.J.C., Chu-Carroll, J.: Statistical source expansion for question answering. PhD thesis, Technologies Institute, School of Computer Science, Carnegie Mellon University (2011)
7. Mehrez, T., Abdelkawy, A., Heikal, Y., Lange, P., Nabil, H., Suendermann-Oeft, D.: Who Discovered the Electron Neutrino? A Telephony-Based Distributed Open-Source Standard-Compliant Spoken Dialog System for Question Answering. In: Proc. of the GSCL, Darmstadt, Germany (2013)
8. Heikal, Y.: An open source question answering system trained on wikipedia dumps adapted to a spoken dialog system. (2013)
9. Greenwood, M.A.: Open-domain question answering. PhD thesis, University of Sheffield, UK (2005)
10. Voorhees, E.M., Harman, D.: Overview of trec 2001. In: Trec. (2001)
11. Ittycheriah, A., Roukos, S.: Ibm's statistical question answering system-trec-11. Technical report, DTIC Document (2006)
12. De Marneffe, M.C., MacCartney, B., Manning, C.D., et al.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC. Volume 6. (2006) 449–454
13. Sonntag, D.: Ontologies and Adaptivity in Dialogue for Question Answering. Volume 4. IOS Press (2010)