QASR: Spoken Question Answering Using Semantic Role Labeling

Svetlana Stenchikova
State University of New York
Stony Brook, NY, 11794
sveta@cs.sunysb.edu*

Dilek Hakkani-Tür **Gokhan Tur**
AT&T Labs - Research
Florham Park, NJ, 07932
dtur,gtur@research.att.com

In this demo, we will present *QASR* - a question answering system that uses semantic role labeling. Given a question, QASR semantically parses it, formulates a query, sends it to a search engine (in this case, Google), and semantically parses the candidate answers matching the semantic argument in question. QASR supports multiple modalities for question answering. It is possible to write down the question using a web or command-prompt interface, or speak the question via speech interface.

Question answering (QA) is the task of finding a concise answer to a Natural Language question. Question answering system uses a search engine but differs from a web search task by the type of its input and output. Input to a search engine is a query while a QA system takes a natural language question as an input. For example, if one wants to find out *"Who first broke the sound barrier?"*, the user of a search engine would convert the question into a query like *first AND broke AND "sound barrier"*, while the user of a QA system would type or say the question. The output of the QA system is a concise answer: *Yeager*, in the case of the above question while the search engine produces snippets with link to the web pages.

In this work we only focus on factoid questions. Answer to a factoid question is a single word or a short phrase. The above example is a factoid question, while *"What is the meaning of life"* is not. Factoid QA has been extensively studied in the framework of the TREC conferences (TREC).

In this work, we apply semantic role labeling to the QA task. Semantic role labeling aims to identify the predicate/argument relations within a sentence. In our approach we process question and candidate sentences extracted using a search engine, identify predicate/argument structure using the *Assert* system (Pradhan et al., 2005) which currently has one of the highest reported performance among other semantic role labelers. Assert assigns tags to predicates and argument phrases: TARGET tag is assigned to the predicate, ARG0, ARG1, ARG2..., ARGM-TMP, ARGM-LOC tags are assigned arguments. The meaning of an argument type depends on a class of the predicate that it appears with, but generally ARG0 serves as an agent and ARG1, ARG2... are objects. Arguments of the type ARGM-TMP which represents temporal argument and ARGM-LOC - a locational argument are shared over all predicates. For example a sentence *"Nostradamus was born in 1503 in the south of France"* gets tagged by the semantic role labeling program: *[ARG1 Nostradamus] was [TARGET born] [ARGM-TMP in 1503] [ARGM-LOC in the south of France].* In this example *born* is identified as a target predicate with three arguments: object *"Nostradamus"*, temporal argument *"in 1503"* and locational argument *"in the south of France"*.

To demonstrate the importance of predi-

*This research has been conducted at AT&T Labs - Research cate/argument extraction for the QA task consider a question *"Who created a comic strip Garfield?"* and a candidate sentence: *"Garfield is a popular comic strip created by Jim Davis featuring the cat Garfield ..."* Without the deep semantic processing and finding predicate/argument relations, we could extract the answer *Jim Davis* by creating an example-specific template. It is not feasible to create templates for each anticipated predicate/answer candidate pair because the number of predicates covered by open-domain question answering system is unlimited as well as the variation of candidate sentences. With the knowledge of the predicate/argument structure identified by the semantic role labeler: *Garfield is [ARG1 a popular comic strip] [TARGET created] [ARG0 by Jim Davis] featuring the cat Garfield ...* ARG0 corresponds to the "agent", the answer to the "who" question is expected to be an agent, so we can extract *Jim Davis* as the answer.

While using the speech interface, QASR uses AT&T Watson speech recognizer (Goffin et al., 2005) with VoiceXML (VXML). The speech recognizer uses trigram language models based on Variable N-gram Stochastic Automata (Riccardi et al., 1996). The acoustic models are subword unit based, with triphone context modeling and variable number of gaussians (4-24). The output of the ASR engine is then used as the input of the SLU component. We train the language models using the questions produced by the TREC conferences and augmented them with questions gathered from the trivia websites. Unlike textual question answering, spoken question answering has not been extensively studied in the QA community, mostly information retrieval, natural language processing, and artificial intelligence communities participated in TREC evaluations.

During the search phase, this demo will need internet access to query the search engine used. In case, there is no internet access, we can show pre-queried questions. Furthermore, to present the speech interface, we will need a phone line. We can always use our cell phones or headphones, but since the acoustic models are tuned to perform optimally on land-line telephones, it would be the best choice.

References

V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tür, A. Ljolje, S. Parthasarathy, M. Rahim, G. Riccardi, and M. Saraclar. 2005. The at&t watson speech recognizer. In *Proceedings of the ICASSP*, Philadelphia, PA, March.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proceedings of the Association for Computational Linguistics 43rd annual meeting (ACL-2005)*.

G. Riccardi, R. Pieraccini, and E. Bocchieri. 1996. Stochastic automata for language modeling. *Computer Speech and Language*, 10:265–293.

Text retrieval conference (TREC). http://trec.nist.gov.

Voice extensible markup language (VoiceXML) version 2.0. http://www.w3.org/TR/voicexml20/.