Desire: Deep Semantic Understanding and Retrieval for Technical Support Services

Abhirut Gupta(✉), Arjun Akula, Gargi Dasgupta, Pooja Aggarwal, and Prateeti Mohapatra

IBM Research, Bangalore 560045, India
{abhirutgupta,arakula2,gaargidasgupta, aggarwal.pooja,pratemoh}@in.ibm.com

Abstract. Technical support services involve enterprises providing after-sales support to users of technology products. The current support structure is labor intensive with practitioners manually consulting support documentation to troubleshoot users' problems. We propose a cognitive technical support system as one that: (a) can understand technical problems expressed by users, (b) can automatically provide relevant resolution information and (c) can learn and improve its understanding and resolution over time. A typical technical problem description contains a combination of symptoms experienced by the user, explanation of attempts already made to resolve the problem, and sometimes, a clear expression of the requirement to solve the problem. Handling such intricate descriptions is outside the scope of current retrieval based systems and requires a deep understanding of the problem, combined with reasoning over a knowledge graph.

Keywords: Knowledge graph · Semantics · Technical support

1 Introduction

Currently, the operating model in technical support services is human-intensive where skilled experts are hired to support users with their queries. The science of support services is aimed at providing high quality support at lower cost by: (a) driving performance optimization from data analytics and (b) reducing manual effort by building cognitive interfaces where machines and human work together. In this paper, we focus on the latter goal of reducing manual effort by building a cognitive technical support system. This support system can interact with users of a technology, understand intricate problem definitions and respond with an accurate answer.

Large IT service providers, having realized the transformation brought in by cognitive technology, are experimenting with cognitive service agents like Amelia [4] and Watson [1] for IT problem resolution. Building blocks of such cognitive technology include components that can automatically parse text (or speech) to reliably perform problem determination, root cause analysis and automated resolution in the domain. There are two non-trivial challenges in building the next generation of cognitive systems: (1) Automatic understanding of problems expressed in natural language and (2) Using that understanding to retrieve correct responses from an Information Retrieval system.

In recent times, a number of classifiers using deep learning techniques have been built for question answering systems [2]. When there are commonly repeating questions, these classifiers perform well. However, the sheer combinations of products, versions and platforms make technical support questions follow a long-tailed distribution [3]. Hence, it is unlikely that two questions will be similar. In this domain, it is thus imperative to semantically understand each question and use the semantics for retrieval. To this effect, knowledge graphs representing entities and relations have become very popular [6,7]. We present Desire(Deep Semantic Understanding and Retrieval for Technical Support Services), a cognitive system for technical support that builds a domain specific knowledge graph and uses deep parsed text fragments for traversing it.

2 Desire Architecture

Figure 1 presents an overview of the Desire system architecture and its main components:

- **Offline Component** Which comprises of - (i) Knowledge Graph Builder that extracts common entities and relations for building a Knowledge Graph (KG), and (ii) Deep Question Understanding that builds a model for recognizing symptom, intent and attempt from user queries expressed in natural language.
- **Online Component** Which comprises of - (i) Query Parser that consults the understanding model to parse out intent, symptom and attempt from the user generated query (ii) Query Builder: Uses the intent, symptom and attempt for semantic traversal of the KG.

We describe these components in detail in the following subsections

2.1 Deep Question Understanding

On analyzing several thousands of problem descriptions, it was observed that troubleshooting queries in technical support often have three distinct parts - `Symptom` (description of the problem), `Attempt` (actions that the user has already tried to resolve the problem), and `Intent` (an explicit request for a certain service) Fig. 2 has an example of all three parts extracted from a user question. Note that some or all of these parts might be present in a given question. Formally, we can formulate the problem of identifying these parts from a problem as follows. Given a problem description as a sequence of words - $T = [t_1, t_2, .., t_n]$, we identify three non-overlapping, possibly empty, subsequences $T_{symp}, T_{att}, T_{int}$ which denote the symptom, attempt and intent respectively.

We use Statistical Information and Relation Extraction (SIRE) [5] for detection of symptom, intent and attempt mentions. The mention detection currently is based on maximum entropy models. It requires a dataset of (hundreds of) manually annotated problem descriptions.

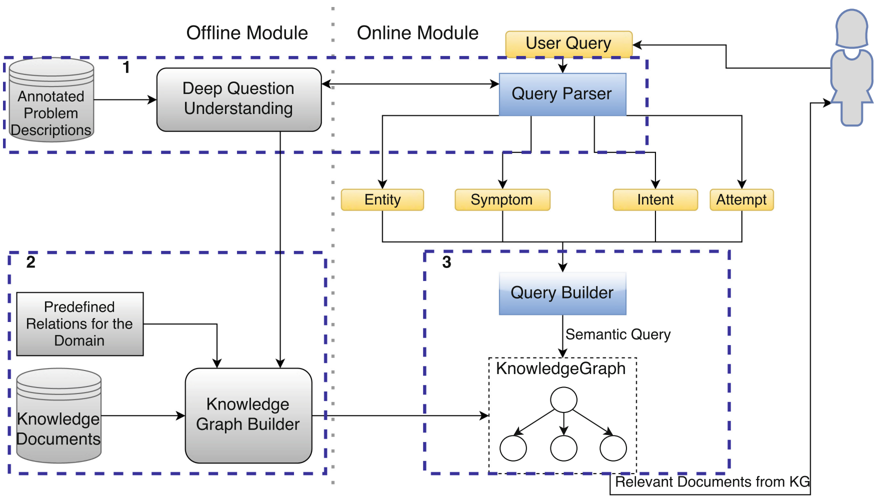
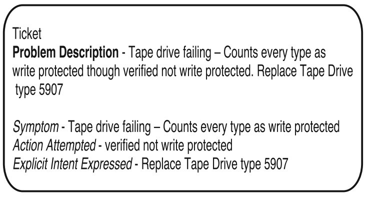


Fig. 1. System architecture with the offline and runtime modules


Ticket
Problem Description - Tape drive failing – Counts every type as write protected though verified not write protected. Replace Tape Drive type 5907

Symptom - Tape drive failing – Counts every type as write protected
Action Attempted - verified not write protected
Explicit Intent Expressed - Replace Tape Drive type 5907

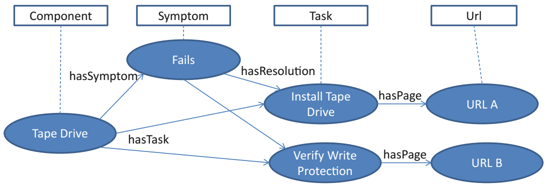


Fig. 2. A sample ticket regarding a failing tape drive

Fig. 3. A section of the Knowledge Graph applicable to the problem.

2.2 Knowledge Graph

A Knowledge Graph is essentially a Knowledge Base that is organized as Entities and the Relations that connect them. Figure 3 shows the KG section that is applicable to the example in Fig. 2. The relations *hasSymptom, hasTask, hasSymptom* and *hasPage* are used to connect entities of the type *Component, Symptom, Task* and *Url*, and are predefined. Instances of these relations are, however, extracted from both the knowledge source documents and by mining frequent patterns in annotated problem descriptions.

Finding the correct resolution documents for a query reduces to a constrained traversal problem in the knowledge graph over the entities and relations. While these queries can be formulated as simple traversals on a KG, they are very difficult for traditional text retrieval systems.

2.3 Query Parser

At runtime, the natural language query that comes to Desire is parsed using the deep question understanding model. The parser extracts out the intent, symptom and attempt from the query.

2.4 Query Builder

The Query Builder utilizes the Relationship Extractor model learnt in the offline module, and translates the parts extracted into a semantic query on the KG. For e.g. the question in Fig. 2 translates to a traversal in the KG as follows - look for all documents for the "Tape Drive" entity and "Fails" symptom where the resolution mentions "replace tape drive" and does not mention "set write protection". Since it is not necessary that all three of intent, symptom and attempt be present in all questions, the module prioritizes different parts of the question in the presence and absence of other parts.

3 Demonstration

We will demo our real-live cognitive technical support system, the Desire system. We have also included a video of our demo in the submission.

References

1. Ante, S.: Ibm set to expand watsons reach. Wall Street J. (2014)
2. Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**(1), 1–127 (2009)
3. Buckley, J.J.: Long tailed distributions. In: Buckley, J.J. (ed.) Fuzzy Probabilities and Fuzzy Sets for Web Planning, pp. 147–159. Springer, Heidelberg (2004). doi:10.1007/978-3-540-36426-9_16
4. Flinders, K.: Meet amelia new artificial intelligence platform interacts like a human. http://www.ipsoft.com/
5. Kambhatla, N., Qian, L., Roukos, S., Sun, Z.: Statistical information and relation extraction (sire). http://researcher.watson.ibm.com/researcher/view_group.php?id=2223
6. Mitchell, T.: Never-ending learning. Technical report, DTIC Document (2010)
7. Pelikánová, Z.: Google knowledge graph (2014)