Information Processing and Management 52 (2016) 340–357

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Genetic programming-based feature learning for question answering

Iman Khodadi, Mohammad Saniee Abadeh*

Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

ARTICLE INFO

Article history:
Received 18 June 2014
Revised 15 September 2015
Accepted 15 September 2015
Available online 10 November 2015

Keywords:
Question Answering (QA)
Feature learning
Genetic Programming (GP) algorithm
Feature weight learning
Factoid questions
Information Extraction (IE)

ABSTRACT

Question Answering (QA) systems are developed to answer human questions. In this paper, we have proposed a framework for answering definitional and factoid questions, enriched by machine learning and evolutionary methods and integrated in a web-based QA system. Our main purpose is to build new features by combining state-of-the-art features with arithmetic operators. To accomplish this goal, we have presented a Genetic Programming (GP)-based approach. The exact GP duty is to find the most promising formulas, made by a set of features and operators, which can accurately rank paragraphs, sentences, and words. We have also developed a QA system in order to test the new features. The input of our system is texts of documents retrieved by a search engine. To answer definitional questions, our system performs paragraph ranking and returns the most related paragraph. Moreover, in order to answer factoid questions, the system evaluates sentences of the filtered paragraphs ranked by the previous module of our framework. After this phase, the system extracts one or more words from the ranked sentences based on a set of hand-made patterns and ranks them to find the final answer. We have used Text Retrieval Conference (TREC) QA track questions, web data, and AQUAINT and AQUAINT-2 datasets for training and testing our system. Results show that the learned features can perform a better ranking in comparison with other evaluation formulas.

1. Introduction

Question Answering systems are advanced search engines that can provide the least brief and the most complete answer to users instead of making them read a set of documents. QA systems are essential tools for dealing with the fast-growing global information. However, upgrading a search engine to a QA system is a complex and open-ended problem (Zadeh, 2003). Machine-based human-like answering has been a dream that Artificial Intelligence (AI) scientists have been trying to achieve. Based on Russell and Norvig (2010), AI field has four definition groups and one of them is based on the Turing test, which is about the ability of machines to communicate or answer like a human. Moreover, Arthur Samuel (Samuel, 1983) in his talk titled "AI: where it has been and where it is going" stated the main goal of AI and machine learning as: "to get machines to exhibit behavior, which if done by humans, would be assumed to involve the use of intelligence."

At the early years, the fundamental problem of QA was converting a natural language question to a Structured Query Language (SQL) query and retrieving answers from structured data. These convert-to-query-based systems have been called restricted-domain systems because they can only answer questions related to their already-provided structured data (Indurkhya & Damerau, 2010). However, with rapid enlargement of data in unstructured format, extracting answers from domain-independent

* Corresponding author. Tel.: +98 2182884349.
E-mail addresses: iman.khodadi@modares.ac.ir (I. Khodadi), saniee@modares.ac.ir (M.S. Abadeh).

http://dx.doi.org/10.1016/j.ipm.2015.09.001
0306-4573/

sources became the main challenge of QA. These QA systems, which operate on sources that could be general and free of specific domain, are called open-domain QA systems. The first web-based QA system developed in 2004, named Start (Katz, Lin, & Felshin, 2002), and contemporary systems are Wolfram[1] from IBM, and AskHERMES (Yen et al., 2013b).

The simplest form of answering for a QA system is returning a paragraph to a definitional question. However, factoid question is the most discussed question type. The answer of a factoid question is a simple fact such as name of a person or a location that can be found in a sentence (Jurafsky & Martin, 2009). In addition to these two question types, there are others such as list, hypothetical, causal, relationship, procedural, and confirmation questions. In this paper, we have worked on definitional and factoid questions with a four-phase framework including paragraph ranking, sentence ranking, word extraction, and word ranking.

In order to select an answer from a set of candidates, they must be ranked. The ranking problem includes computing a score based on a set of features. Computing the sum of all feature values or $\sum_i (feature_i)$ formula cannot reflect the true value of an answer because each feature has a weight. Finding the weights is a supervised learning problem that can be solved by a discriminant-based classification algorithm. In this paper, we used three methods for this task including Linear Discriminant Analysis (LDA), Logistic regression, and Support Vector Machines (SVM). Taking into account the weights, the score computation formula will become $\sum_i (feature_i \times weight_i)$.

Following these two formulas, one possible continuation is using other arithmetic operators such as multiplication, division, exponential, and logarithm, which is the main purpose of this paper. Here a challenging problem is finding a promising ranking formula based on a set of operators and features. In order to solve such kind of problems, evolutionary approaches can be used since they are capable of searching in large-scale search spaces effectively. Among different types of evolutionary algorithms, Genetic Programming (GP) whose individuals are trees would be the best candidate. Since our problem is finding a ranking formula, which can be modeled as a tree of operators and features, GP would be a perfect choice. The main contribution of this paper is learning efficient features via GP algorithm for a Question Answering system.

The remainder of this paper is organized as follows: in Section 2, we will discuss about related works. In Section 3, the structure of our proposed QA system will be presented in detail. Sections 4 and 5 deal with answering definitional and factoid questions. The last three sections describe experimental results, discussion, and conclusions respectively.

2. Related works

Question answering subject is discussed by chapter books in Indurkhya and Damerau (2010), and Jurafsky and Martin (2009), and by survey papers such as Kolomiyets and Moens (2011).

Previous works on feature engineering task includes Severyn and Moschitti (2013), Severyn et al. (2013), Tymoshenko, Moschitti, and Severyn (2014), and Severyn and Moschitti (2012). Tymoshenko et al. (2014) represented question and candidate answer passages with pairs of shallow syntactic/semantic trees whose constituents are connected using Linked Open Data (LOD). The trees are processed by SVM and tree kernels, which can automatically exploit tree fragments. Severyn and Moschitti (2013) used automatic feature engineering for answer-selection task as an alternative to manual rule definition. Severyn, Nicosia, and Moschitti (2013) proposed a method to learn automatically complex patterns such as relational semantic structures occur in questions and their answer passages. They achieved this task by providing their learning algorithm with the trees derived from the syntactic trees of questions and passages connected by relational tags, where the latter are provided by automatic classifiers. Severyn and Moschitti (2012) defined a novel supervised approach that exploits structural relationships between a question and its candidate passages to learn a re-ranking model. They encoded structures in SVM by means of sequence and tree kernels, which can implicitly represent question and answer pairs in huge feature spaces.

State-of-the-art text-ranking papers include Yen et al. (2013a), Moschitti and Quarteroni (2011), Heie et al. (2012), Moreda et al. (2011), and Ko et al. (2010). Yen et al. (2013a) proposed an SVM-based model for context ranking, which focuses on weighting the Named Entities (NEs) terms based on their contextual clues such as position information, Named-Entity words, phrase structures, and question terms. They combined rich features to predict whether the input content is relevant to the question type. Moschitti and Quarteroni (2011) explored linguistic kernels for answer re-ranking. They used supervised discriminative models that learn to rank answers using examples of question and answer pairs. They used four features: bag-of-words, *N*-grams, syntactic chunks, and head noun phrase-verb phrase-prepositional phrase (NP-VP-PP) groups. Heie et al. (2012) proposed a statistical language modeling-based ranking using keywords and question type features, and a mathematical model for answer extraction. Their results show that the best performance is achieved when they are using web data and exploiting data redundancy. Using semantic features in QA is discussed in Moreda, Llorens, Saquete, and Palomar (2011). They investigated the influence of using two semantic-based features: semantic roles and WordNet, for the answer extraction module of a general open-domain QA system. Ko et al. (2010) proposed a unified probabilistic framework, which combines multiple evidences to address challenges in answer ranking and answer merging. Their framework combines answer relevance and similarity features, and uses five extractors for evaluation including finite state transducers and SVM.

Web-based QA systems generally perform a big-domain search in web data and evolutionary algorithms can be employed in the main structure of these systems. Evolutionary methods with web-based QA systems are discussed in Khodadi and Saniee (2014), Atkinson, Figueroa, and Andrade (2013), and Figueroa and Neumann (2008). In Khodadi and Saniee (2014), we used

[1] www.wolframalpha.com