Ontology Matching with Word2Vec

Harmen Prins

February 16, 2016

Summary

Contents

1 Introduction

2 Problem

2.1 Short Context

On the Semantic Web, ontologies are used to define and share knowledge.

2.2 Problem

When knowledge bases are combined, their ontologies need to be merged.

2.3 Research questions

2.4 Hypotheses

2.5 Justification

Reliable, error proof ontology matching will be key for the distributed Semantic Web.

3 Context

In this section I will discuss all concepts that are required to understand the study.

3.1 The Semantic Web

The Semantic Web, or Web 3.0, is a concept of a Web that enhances the User Experience in ways that the current Web can not provide. Web 2.0, the current Web, models the internet as *hubs* (or sites) with a specific purpose and barely any interconnectivity. For example, Facebook is a hub for sharing content with people you consider friends, but if you want to share that content with more people you will have to go to another hub, like imgur or Pinterest. The Semantic Web, on the other hand, sees web pages as sources of *information* that can be combined whenever the user needs it. This allows not only humans to browse the internet, but also Artificial Intelligences, which will use this information to aid their users.

For example, if a user wants to see a movie this weekend, he orders his AI, or *softbot*, to find him a suitable time and movie to see. The softbot then accesses the Internet Movie Database to see which movies currently play and which of those are similar to what the user liked in the past. He will then access the pages of the local movie theaters to see when those movies play. Lastly, he will contact the softbots of the friends of the user to ask if they will also come to see that movie. After a short exchange the optimal time, place and participant set are decided, and the softbot tells the user what the results are. When the user accepts, the appointment is automatically added to his agenda.

All of this is possible, if the softbot can access all of the information reliably. In the current Web, this is not possible, as all information is written in an ambiguous, unstructured format called human language. To allow softbots to access the same information that we can, the Semantic Web proposes to add a layer to the internet where all information is saved in a unified format that is unambiguous, robot-interpretable and can be used to store any type of knowledge.

3.1.1 Goal

The goal of the Semantic Web is very pragmatic: help people in everyday activities by leveraging all information available on the internet. The fact that it is pragmatic goes a long way of making it a reality: if even one application comes into existence that uses online information in a structured way and thus makes everyday tasks a little easier, the goal is accomplished. Of course, then the goal resets and we should make more applications leveraging more information for more tasks.

3.1.2 Current state

3.1.3 Technologies

3.2 Ontologies

3.2.1 OWL

3.2.2 Role

3.3 Current merging strategies

Many different merging strategies have already been developed. All strategies follow the same two-step approach. The two steps are independent and as such different methods can be used interchangeably. The first step is to generate an initial correspondence set, where correspondences between nodes are found based on just their labels and meta-data. The second step is to use this initial set and the structure of the ontology to find more correspondences. These steps are called the terminological and the structural steps. Some strategies also use extra steps like the extensional and semantic steps, which use vector space models and inference, respectively. https://hal.inria.fr/hal-00917910/document

Popular terminological strategies are WordNet comparison and edit distance. The former uses the popular WordNet hierarchy as a distance measure between to concepts, e.g. the number of edges between the concepts and their least general common superconcept. Edit distance is a purely string-based similarity measure. The similarity is a (weighted) count of edits required to transform one word into another, where common edits are insertion, deletion and replacement. Similar methods include substring matching and n-gram matching, which compare parts of the strings to find similarities.

3.4 Other related research

3.4.1 Word sense representation

Skip-gram models have been widely used to represent words as vectors. This is useful as it is easy to calculate the distance between two vectors, which is a useful property for ontology matching. This vector distance has been shown to be correlated with the semantic distance between words. Recently, an effective and efficient skip-gram model has been developed. It is called word2vec, and uses a number of extensions over previous methods that enable it to efficiently learn an effective representation.

Word2vec learns this representation by trying to predict a context matrix. First a one-hot encoding is made for the vocabulary. Then the model learns to predict a context matrix of C one-hot vectors based on the given one-hot input vector. The model is a neural network with one hidden layer and multiple output layers. Since the words are one-hot encoded the input to the hidden layer is equal to the row of the weight matrix that corresponds to the word. This row is the vector representation of the word.

3.5 Used technology

3.5.1 Ontology constructor

3.5.2 AgreementMaker

AgreementMaker is an ontology matching system which obtained the highest F-measure in 6 of the 7 ontology matching tracks of OAEI 2015

The system uses an extensible framework which allows its users to add new modules to the system. This is very useful for research as it allows the researcher to compare different modules by swapping in just those modules in the framework. The module-swapping technique will also be used in this work, but more of that will be discussed in *Evaluationmethods*.