Knowledge Tagger: Customizable Semantic Entity Resolution using Ontological Evidence

Panos Alexopoulos, Boris Villazon-Terrazas, and José-Manuel Gómez-Pérez

iSOCO, Avda del Partenon 16-18, 28042, Madrid, Spain,
{palexopoulos,bvillazon,jmgomez}@isoco.com

Abstract. Knowledge Tagger performs Named Entity Resolution (NER) in texts using relevant domain ontologies and semantic data as background knowledge. Its distinguishing characteristic is its disambiguation-related customization capabilities as it allows users to define and apply custom disambiguation evidence models, based on their knowledge about the domain(s) and expected content of the texts to be analyzed. In this demo we explain the structure and content of such evidence models and we demonstrate how, given a concrete resolution scenario, one may use our system to define and apply them to texts pertaining to this scenario.

1 Introduction

In this paper we demonstrate Knowledge Tagger[1], a system that utilizes background semantic information, typically in the form of Linked Data, to accurately determine the intended meaning of detected semantic entity references within texts. The system is based on a novel corresponding framework [1] that we have developed and which is particularly applicable to constrained scenarios where knowledge about what entities and relations are expected to be present in the texts to be analyzed is available.

More specifically, through a structured semi-automatic process the framework enables i) the exploitation of this a priori knowledge for the selection of the subset of domain semantic information that is optimal for the disambiguation scenario at hand, ii) the use of this subset for the generation of corresponding evidence and iii) the use of this evidence for the disambiguation of entities within the scenario's texts. As we have already shown in [1] this process allows our system to adapt to the particular characteristics of different domains and scenarios and be more effective than other similar systems primarily designed to work in open domain and unconstrained scenarios like, for example, DBPedia Spotlight [3], AIDA [2] or the systems included in NERD [4].

2 Framework and System Overview

Knowledge Tagger's underlying framework is based on the intuition that a given ontological entity is more likely to represent the meaning of an ambiguous term when there are many ontologically related to it entities in the text. The latter can be seen as

[1] http://glocal.isoco.net/disambiguator/demo

evidence whose quantitative and qualitative characteristics can be used to determine the most probable meaning of the term. Nevertheless, which entities and to what extent should serve as evidence in a given scenario depends on the domain and expected content of the texts that are to be analyzed. For that, the key ability our system provides to its users is to construct and use, in a semi-automatic manner, custom ontology-based disambiguation evidence models.

Such models define for given ontology entities which other entities and to what extent should be used as evidence towards their correct meaning interpretation (see Table 1). Their construction depends on the characteristics of the domain and the texts. For example, assume we want to disambiguate location references within textual descriptions of military conflicts like the following: *"Siege of Tripolitsa occured near Tripoli with Theodoros Kolokotronis being the leader of the Greeks against Turkey"*. The nature of these texts allows us to expect to find in them, among others, military conflicts, locations where these conflicts took place and people and groups that participated in them. This in turn allows us to use these entities as evidence for disambiguating one another. For example, in the above text the term "Tripoli" is mentioned along with terms like "Siege of Tripolitsa" (a battle that took place in Tripoli, Greece) and "Theodoros Kolokotronis" (the commander of the Greeks in this siege). Thus, it is fair to assume that this term refers to the Greek town of Tripoli rather than, for example, to Tripoli of Libya. Generalizing this, we may define the location disambiguation evidence model of Table 2 where, for instance, a populated place can be disambiguated by the military conflicts that took place in it (row 1) and by the military persons that fought in conflicts that took place in it (row 3).

Table 1. Examples of Target-Evidential Entity Pairs for the Miltary Conflict Scenario

Location	**Evidential Entity**	**dem**
dbpedia:Columbus,_Georgia	James H. Wilson	1.0
dbpedia:Columbus,_New Mexico	dbpedia:Pancho_Villa	1.0
dbpedia:Beaufort_County,_South_Carolina	dbpedia:James_Montgomery_(colonel)	0.25
dbpedia:Beaufort_County,_North_Carolina	dbpedia:John_G._Foster	1.0

Table 2. Sample Disambiguation Evidence Model for Military Conflict Texts

Target Concept	**Evidence Concept**	**Relation(s) linking Evidence to Target**
dbpedia-owl:PopulatedPlace	dbpedia-owl:MilitaryConflict	dbpprop:place
dbpedia-owl:PopulatedPlace	dbpedia-owl:MilitaryConflict	dbpprop:place, dbpedia-owl:isPartOf
dbpedia-owl:PopulatedPlace	dbpedia-owl:MilitaryPerson	is dbpprop:commander of, dbpprop:place
dbpedia-owl:PopulatedPlace	dbpedia-owl:PopulatedPlace	dbpedia-owl:isPartOf

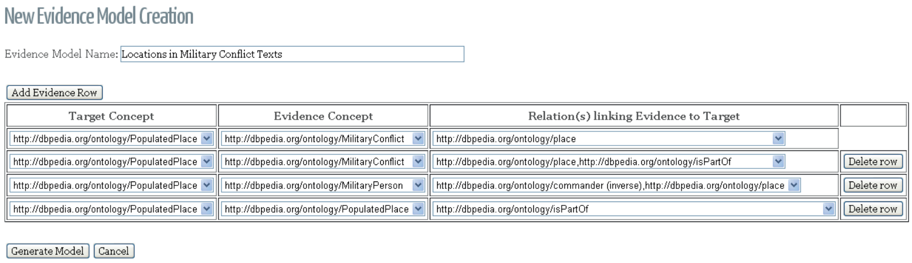

Fig. 1. New Evidence Model Creation Form

To define this model in the Knowledge Tagger demo we work as follows. First we press the **“Create New Evidence Model”** button to reveal the model creation form. Then we give a name for the new model (e.g. “Locations in Military Conflict Texts”) and we start filling the table form with the information of Table 2 (see Figure 1). First we select the target concept (e.g. “PopulatedPlace”), then the one to be used as evidence (e.g. “MilitaryConflict”) and then the (automatically calculated) relation path between them that we want to consider. For simplicity, in this demo we consider paths of maximum length two.

When the model is complete we press the **“Generate Model”** button to store the model into the server and generate target-evidence entity pairs. Each pair is accompanied by a degree that quantifies the evidential entity’s strength for the given target. (see table 1). For example, James Montgomery acts as evidence for the disambiguation of Beaufort County, South Carolina because he’s fought a battle there while his evidential power for that location is 0.25, practically because there are 3 other military persons in the ontology also named Montgomery. The exact way this strength is calculated may be found in [1]. In any case, depending on the size of the underlying ontology, the generation of the target-evidence pairs can take a while but it’s a process that will need to be performed only once. For this example, the creation of the model takes about 30 seconds in a standard server environment.

When the generation process is finished, the new model appears as an option in the list of defined evidence models and can be used to perform entity detection and disambiguation. To do that we select the model and then use the ”Input Text” form to perform NER to texts relevant to the scenario the model has been defined for. By pressing the **“Perform NER”** button the system works as follows: First it extracts from the text terms that possibly refer to the target entities as well as those that refer to their respective evidential entities. Then the disambiguation evidence model is used to compute for each extracted term the confidence that it refers to a particular target entity. The target entity with the highest confidence is expected to be the correct one. Figure 2 shows the results of executing this process on the above text about Siege of Tripolitsa.

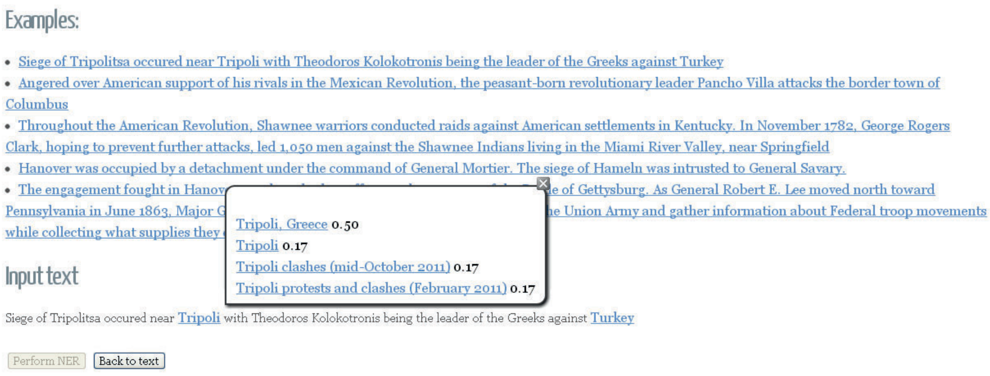

Fig. 2. Semantic Entity Resolution Example

3 Conclusions and Future Work

Knowledge Tagger does not aim to be independent of the content or domain of the input texts but rather adaptable to them. That's exactly its main differentiating feature from other similar systems as our purpose was not to build another generic disambiguation system but rather a reusable framework that can be adapted to the particular characteristics of the domain and application scenario at hand and exploit them to increase the task's effectiveness.

The current version of the system's user interface is still in an early stage of development. A first line of future work will focus on adding more domain knowledge to the system's repository (other than the football and history datasets we already have) so that users are able to build evidence models for a larger range of domains. Moreover, we intend to allow users to use their own semantic data by linking our system to their repository.

References

1. Alexopoulos, P., Ruiz, C., Gomez-Perez, J.M.: Scenario-Driven Selection and Exploitation of Semantic Data for Optimal Named Entity Disambiguation. In Proceedings of the Semantic Web and Information Extraction Workshop (SWAIE 2012), Galway, Ireland, October 8-12, 2012.
2. Hoffart, J., Yosef, M.A., Bordino, I., Frstenau, H, Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 782-792.
3. Mendes, P.N., Jakob, M., Garcia-Silva, A., Bizer, C.: DBpedia spotlight: shedding light on the web of documents. In Proceedings of the 7th International Conference on Semantic Systems, ACM, New York, USA, 1-8, 2011.
4. Rizzo G., Troncy, R.: NERD: A Framework for Evaluating Named Entity Recognition Tools in the Web of Data. In 10th International Semantic Web Conference, Demo Session, pages 1-4, Bonn, Germany, 2011.