Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference

Brochette: Toward A Continuous Learning Platform for Knowledge Acquisition and Integration

Ivan Berlocher, Kim Seonho, Tony Lee
Saltlux Inc, Daewoong Bldg. 689-4, Yeoksam 1 dong, Gangnam-gu, Seoul, South Korea
{ivan, shkim, tony}@saltlux.com

Abstract

Question-Answering (QA) systems like IBM Watson are particularly challenging to design and need to cover areas including computational linguistics, information retrieval, knowledge representation and reasoning, and machine learning. 'Exobrain' is a Korean Research Program which aims at building such a high-performing QA system for the Korean Language. In this paper, we describe 'Brochette', a continuous learning platform that iteratively acquires a large volume of unstructured data and extracts sentences related to entries of an ontology, a formal representation of knowledge used for further queries and reasoning purposes. As various Machine Learning modules must be trained and tuned to suit new content and adapt to new domains, Brochette platform also provides a framework for continuously re-training and evaluating machine learning components for syntactic and semantic analysis. Integrating semantic information from the ontology and results of these machine learners, the system is able to discover new knowledge which is then incorporated again in the Knowledge Base, making it continuously evolving.

Introduction

We describe Brochette, a system for semi-automatically acquiring knowledge from online sources like encyclopedias or dictionaries and producing semantic annotation corpora for further various machine learners. These machine learning agents are able to discover new facts and entities, populating a very large Knowledge Base dedicated to enlarging the coverage and precision of a high performance Question-Answering System for the Korean Language.

Our work is inspired by NELL (Carlson et al. 2010), a system which acquires two types of knowledge:

(1) knowledge about which noun phrases refer to which specified semantic categories, such as cities, companies, and sports teams, and
(2) knowledge about which pairs of noun phrases satisfy which specified semantic relations, such as hasOfficesIn(organization, location).

We have the same goal as NELL in term of self-learning for populating a Knowledge Base (KB), but our approach is radically different. In NELL, four main components are tightly designed for learning facts about the two types of knowledge describe above, whereas our platform is a generic framework which can plug in any type of annotator. The annotators can be cascaded in customizable workflows to learn new knowledge. The Brochette ontology is expressed in RDF, whereas the NELL ontology is not.

Input Ontology

The Knowledge Base is an ontology called "XB Ontology" and the schema (classes and properties) has been built manually from analysis of various resources like Wikipedia, YAGO (Suchanek, Kasneci and Weikum, 2007), DBpedia and KorLex, the Korean WordNet from Busan University (Yoon, Hwang, Lee and Kwon, 2009), which contains mapping information to the original Wordnet synonyms set. To define classes, at first, we got the intersection of KorLex nouns and YAGO classes. KorLex includes 100,000 nouns that are sufficient to describe knowledge in Korean, while YAGO classes fully cover the instances of Wikipedia. Because both YAGO and KorLex have mapping information to WordNet, calculating intersection of KorLex nouns and YAGO classes is straightforward. As a result, 5,775 classes were derived. To simplify the integration of NLP results to the ontology the Tag set of a Named Entities Recognizer (NER) was also used. Similarly as YAGO (Suchanek, Kasneci and Weikum, 2007) semi-structure information of Wikipedia, particularly info-boxes and categories have been used to make rules for extracting automatically instances rules Wikipedia (the Korean version) as shown in Figure 1.

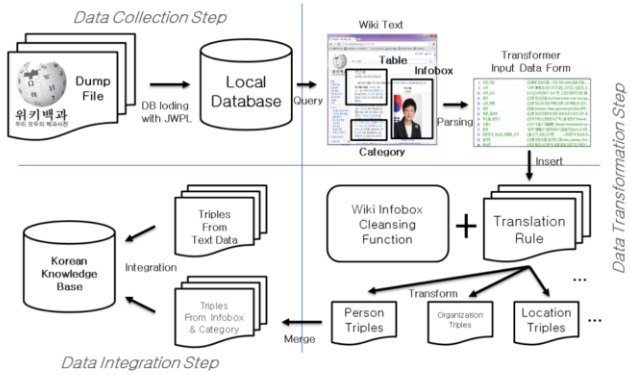


Figure 2: Automatic Instances creation from Wikipedia

For the properties, from the analysis of Wikipedia info-box attributes, YAGO properties and DBPedia properties, 129 properties have been first defined. As semi-automatically constructed ontology from Wikipedia semi-structured information has limitations both in terms of coverage and precision, human curators are needed to check the results of automatically extracted ontology, had new classes, instances or properties base on the text in Wikipedia main page. A web based semantic annotation tool allows curators to semantically annotate documents in RDFa and edit (add, remove, modify) the ontology. At the current state of this project, the XB Ontolgy has is 6,116 classes with 583 properties, 229,654 instances resulting in 1,632,861 triples.

Brochette System Architecture

One key requirement for a successful QA is to acquire high quality corpora from which new knowledge can be extracted. Despite the breadth of knowledge on the Web, experiments (Schlaefer et al., 2011) show that QA performance does not necessarily improve or may even degrade if sources are indiscriminately added. We follow their source acquisition procedure, which is an iterative development process of acquiring new collections of documents to cover salient topics. The design of the system is depicted in Figure 2.

Sources Acquisition

In the case of Watson (Schlaefer et al. 2011), acquiring general-purpose title-oriented document sources such as Wikipedia has been shown to improve QA performance, so we followed this approach for our platform.

In order to acquire knowledge we designed a Fetcher that can crawl News and Blogs from the Web in real time, a Batch Downloader that downloads sources like Wikipedia or Wiktionary and a Meta-Search engine that crawls documents from top results of various search engines like Google or Yahoo! corresponding to a given query. This component is intended to find documents related to a given entity in cases where the previous corpus does not contain enough documents. It is also used to find related or similar documents for a given document to expand the knowledge base. The documents are then indexed by a local search engine for fast retrieval of passages, sentences or documents related to a given topic.

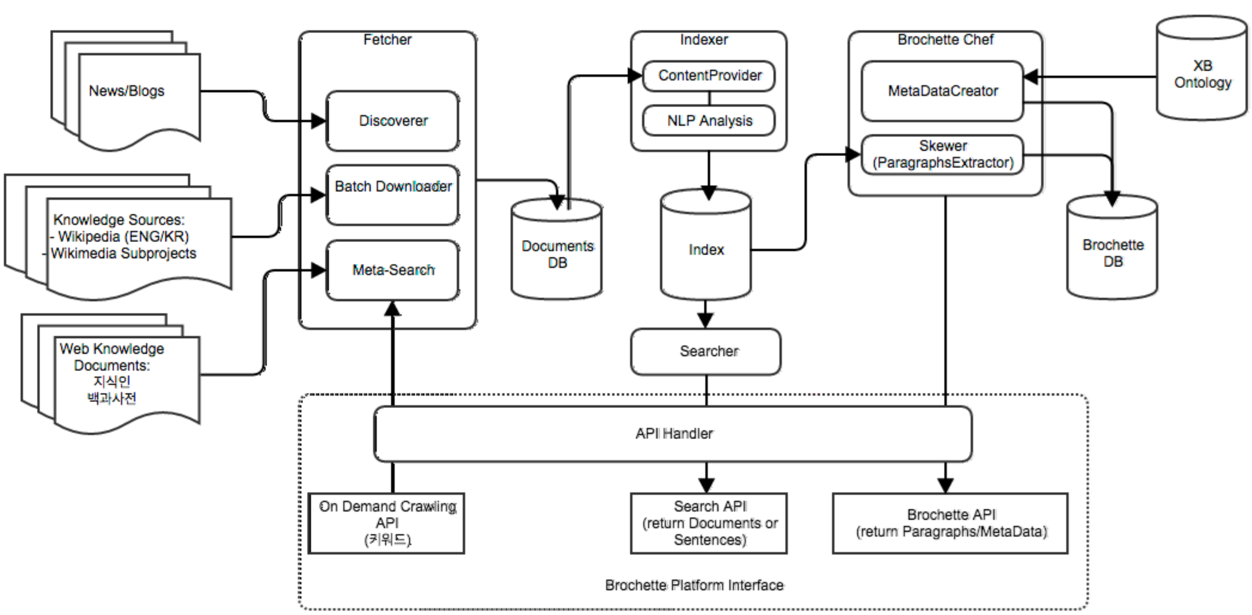
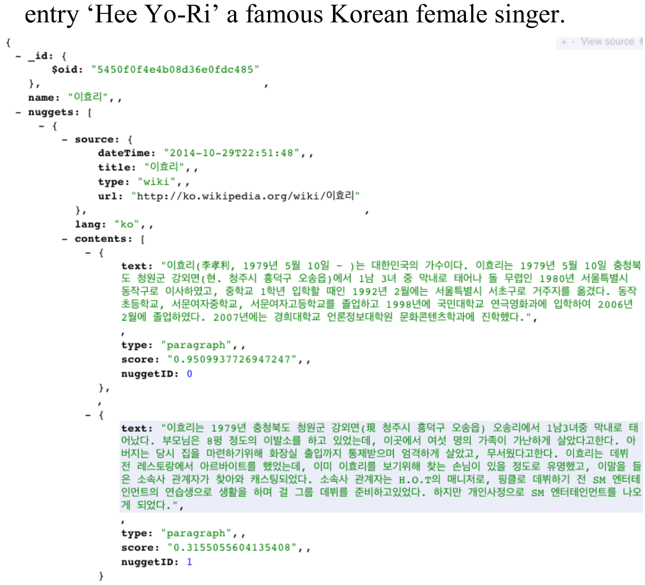


Figure 1: Brochette System Overview

Ontology Integration

We designed a component called 'Brochette Chef' that produces a set of sentences semantically related to an instance of the 'XB Ontology'. The Brochette Chef module comprises 2 components: the Metadata Creator and the Skewer. The Metadata Creator iterates over each entity of the ontology and gets semantic metadata like class information (Person, Company etc.), synonyms, related entities etc. from which it makes a vector of features capturing the semantic of the entities. Our approach follows the DBpedia spotlight (Mendes et al., 2011) algorithm. The entity of the ontology has a description page related to a dictionary or encyclopedia like Wikipedia, so the full text is analyzed, removing common nouns, stop-words, and words in hyper-links, categories or info-boxes are kept. As explained by Mendes, the weighting of features uses the TF (Term Frequency) with ponderation by the ICF (Inverse Candidate Frequency). With this vector representation (Vector Space Model) of each entity of the KB, the semantic of the entity is kept so we call it the 'semantic vector' and it serves as a basis for later semantic matching of entities in text.

The skewer uses the semantic vector related to one KB entry to get the top documents related to the entry from the internal search engine then splits each document into paragraphs and sentences. For each sentence or paragraph, a similarity between the semantic vector of the ontology instance and the features vector of the sentence is calculated and only the sentences or paragraphs with a high threshold similarity are kept. The set of all sentences is stored as a pseudo-document, called a 'brochette'. The figure 3 below illustrates an example of a brochette for the entry 'Hee Yo-Ri' a famous Korean female singer.



```
{                                                    + - View source
 - _id: {
      $oid: "5450f0f4e4b08d36e0fdc485"
   },                                          ,
   name: "이효리",,
 - nuggets: [
    - {
       - source: {
            dateTime: "2014-10-29T22:51:48",,
            title: "이효리",,
            type: "wiki",,
            url: "http://ko.wikipedia.org/wiki/이효리"
         },                                                    ,
         lang: "ko",,
       - contents: [
          - {
               text: "이효리(李孝利, 1979년 5월 10일 ~ )는 대한민국의 가수이다. 이효리는 1979년 5월 10일 충청북도 청원군 강외면(현. 청주시 흥덕구 오송읍)에서 1남 3녀 중 막내로 태어나 돌 무렵인 1980년 서울특별시 동작구로 이사하였고, 중학교 1학년 입학할 때인 1992년 2월에는 서울특별시 서초구로 거주지를 옮겼다. 동작초등학교, 서문여자중학교, 서문여자고등학교를 졸업하고 1998년에 국민대학교 연극영화과에 입학하여 2006년 2월에 졸업하였다. 2007년에는 경희대학교 언론정보대학원 문화콘텐츠학과에 진학했다.",
               ,
               type: "paragraph",,
               score: "0.9509937726947247",,
               nuggetID: 0
            },
            ,
          - {
               text: "이효리는 1979년 충청북도 청원군 강외면(現 청주시 흥덕구 오송읍) 오송리에서 1남3녀중 막내로 태어났다. 부모님은 8평 정도의 이발소를 하고 있었는데, 이곳에서 여섯 명의 가족이 가난하게 살았다고한다. 아버지는 당시 집을 마련하기위해 화장실 출입까지 통제받으며 엄격하게 살았고, 무서웠다고한다. 이효리는 데뷔 전 레스토랑에서 아르바이트를 했었는데, 이미 이효리를 보기위해 찾는 손님이 있을 정도로 유명했고, 이말을 들은 소속사 관계자가 찾아와 캐스팅되었다. 소속사 관계자는 H.O.T의 매니저로, 핑클로 데뷔하기 전 SM 엔터테인먼트의 연습생으로 생활을 하며 걸 그룹 데뷔를 준비하고있었다. 하지만 개인사정으로 SM 엔터테인먼트를 나오게 되었다.",
               ,
               type: "paragraph",,
               score: "0.3155055604135408",,
               nuggetID: 1
            }
```


Figure 4: Example of a brochette in JSON format

For analyzing each sentence, several Natural Language Processing (NLP) components are used, such as Part-of-speech (POS) Tagger, Named Entities Recognizer (NER) and dependency parser. All these modules came from different organizations, are implemented in several languages like C++ or Java, have various output formats and specific settings for executing them. In such an environment, we developed a 'Continuous Learning Framework' based on UIMA (Ferrucci and Lally, 2004) for standardizing the input/output of modules using CAS, the Common Analysis Structure for the annotations produced, and UIMA-AS (Asynchronous Scaleout) for supporting distributed computing.

Continuous Learning Framework

Annotator Engines (AE) in the UIMA terminology sense, like NER or Tagger, are Machine Learning based algorithms and they should be able to be re-trained online given feedback about incorrectly tagged results.

For training annotators a training set has to be created; to be evaluated a test set has to be provided too. This is the role of the Corpus Manager, Evaluations Handler and

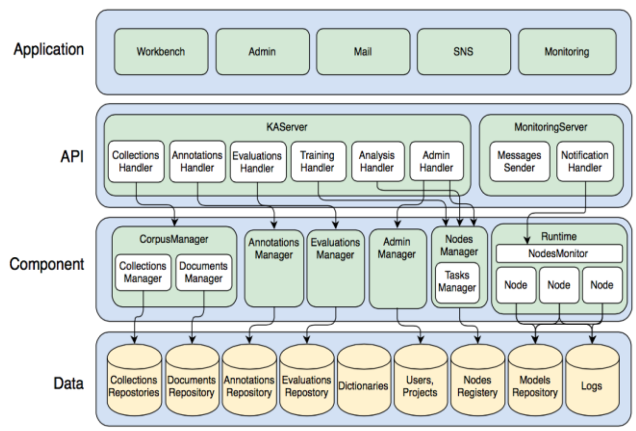


Figure 3: Continuous Learning Framework

Training Handler. The performance of each annotator has to be monitored over time assuring that the precision and recall of the overall system is increasing. This functionality is assured by the Evaluations Manager. The framework also provides a pluggable mechanism allowing adding or interchanging any other Annotator. We implemented a component called a 'Node' providing a Server/Client wrapping of the core NLP engines with a common interface incorporating four main methods: train, load, evaluate, analyze. The result of each engine is encapsulated in a CAS container provided by the UIMA framework. The framework provides a Graphical User Interface (GUI), shown in figure 5, through which human curators can easily:

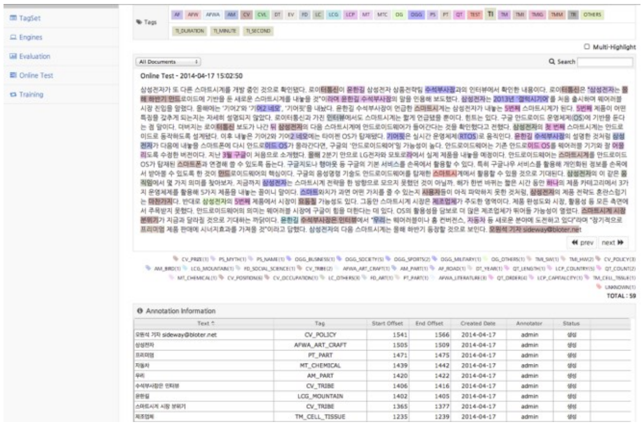


Figure 5: Annotations results UX

- add/remove a Document to a training/testing set
- register new Annotators Engines
- train/evaluate an Annotator
- test a given Annotator for a given text
- correct a badly tagged Document
- add/remove Resource to Dictionaries

Ultimately, the system generates new entities or facts not explicitly included in the input data, so that Annotator Engines have a better chance of analyzing them correctly, and increasing the coverage of the KB.

Experimental Evaluation

The starting corpus was the Korean version of Wikipedia containing 287,466 entries mapped to the XB Ontology.

By applying source acquisition and expansion, adding several famous Korean online encyclopedias, the Encyclopedia Britannica and nearly 2000 further dictionaries including Wiktionary, the number of documents rose to just under 3,000,000 documents, nearly ten times the size of the original corpus. Like Schlaefer et al. (2011), we cannot at this stage measure the impact of the project on the accuracy and precision of the full QA system, but through the brochettes (sets of sentences, semantically related to an instance of the ontology), human curators were able to triple the speed of discovery of new entities or facts compared to the previous approach of annotating only Wikipedia pages.

It shows that the semantic matching proposes a high precision relation of sentences to given entities, while retaining the discovery of new facts about this entity. Further experiments are still needed to quantify the precision recall of this method. That can be improved using more advanced features and maybe also using a machine learner approach.

Conclusion

In this paper, we described a platform supporting continuous acquisition and seamless integration of knowledge from various sources and able to integrate various Annotator Engines for extracting new knowledge from unstructured data into a formal Knowledge Base helping to boost quality of the overall QA system.

This is an ongoing work and it is still early to evaluate it in terms of precision / recall or coverage of the full QA system itself, but such a platform definitively boost the productivity of manual tasks for human curators and helps engineers to visualize the performance of their engines. Further work will be done to quantify the quality of suggestion of new knowledge (entities or facts) and coverage of QA system for a given question set.

Acknowledgements

This work was supported by the Industrial Strategic Technology Development Program (10044494, WiseKB: Big data based self-evolving knowledge base and reasoning platform) funded by the Ministry of Science, ICT & Future Planning (MSIP, Korea)

References

N. Schlaefer, J. Chu-Carroll, E. Nyberg, J. Fan, W. Zadrozny, and D. Ferrucci. Statistical source expansion for question answering, in Proceedings of the 20th ACM International Conference on Information and Knowledge Management, ACM, New York, NY, 2011.

D. Ferrucci and A. Lally. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Natural Language Engineering, 10(3-4):327–348, 2004.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. : Building Watson: An Overview of the DeepQA Project. AI Mag, Vol 31 (2010) 59–79

F. M. Suchanek, G. Kasneci and G. Weikum. : YAGO: A Core of Semantic Knowledge. Proceedings of the 16th international conference on World Wide Web. ACM New York (2007) 697-706.

Mendes et al., DBpedia spotlight: shedding light on the web of documents, Proceedings of the 7th International Conference on Semantics Systems. ACM(2011).

Aesun Yoon, Soonhee Hwang, Eunroung Lee, and Hyuk-Chul Kwon. : Construction of Korean wordnet KorLex 1.5. Journal of KIISE: Software and Applications, Vol 36(1). (2009) 92–108.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an Architecture for Never-Ending Language Learning. In AAAI.