Knowledge graph construction for research literatures

Alisson Oldoni

A research project report submitted for
the degree of
Master of Computing and Information Technology

School of Computer Science and Engineering
The University of New South Wales

14 December 2011

Abstract

Contents

Chapter 1

Introduction

A Knowledge Graph (KG), also known as the knowledge base, is a collection of the machine-readable database that contains entities, the attributes of entities and the relationships between entities [6]. It is an essential foundation for many applications that requires machine understanding.

Popular search engines such as Google [6] and Bing [2] are all leveraging Knowledge Graphs as to provide entity summary information and the related entities based on the query that the user is searching for.

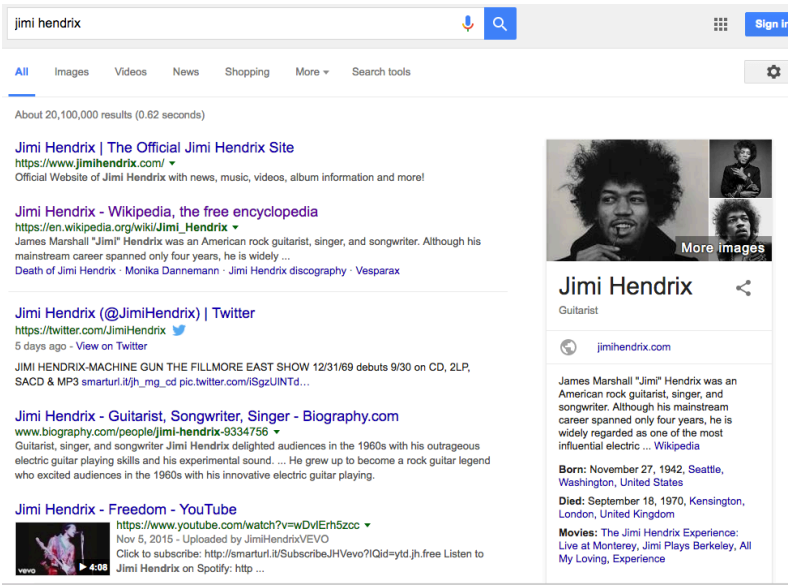

Figure 1.1: An example of knowledge graph application in the Google's result page.

The use of Knowledge Graphs then allow users to be able to see extra information in a summarised table-like form, as to resolve their query without having to navigate to other sites. Note in the example in Figure 1.1

how the right column represents a sequence of facts of the *Jimi Hendrix* entity, in this case an entity of the class (or type) *person*, such as: his official website; where and when he was born; where and when he died; and a list of movies where this person is the subject of.

In this project, we focus on building a domain-specific knowledge graph from research literatures. More specifically, we focus on papers from the topic of databases and attempt to extract information from these papers in order to build a knowledge graph. The first step to achieve in achieving our goal is obtaining the raw natural language text from papers n the area. Most research products such as thesis, papers, or any other report, are mostly available primarily in the Portable Document Format (PDF) format [1] - which then needs to then be parsed into a raw text in an automated manner. Once this text is available, it is then parsed and processed into the following workflow:

1. Discover entities in the text;
2. Discover relationships between these entities;
3. Design effective and efficient algorithms to extract entities and relationships;
4. Perform entity disambiguation and linking to a reference Knowledge Graph (e.g., Yago2 or DBPedia).
5. Improve the quality of the output via input data cleaning, robust extraction, and learning-based post-processing methods;
6. Presenting the facts in a graph (the Knowledge Graph).

In the next sections, this document will give some background information on the techniques needed to achieve the above (Chapter 2), and it will also define the problem more precisely (Chapter 3), building as to introduce the development of this research (Chapter 4). In Chapter 5 we will describe some of the results, followed by the some final remarks in Chapter 6.

Chapter 2

Information Extraction from Natural Language Text

In this project, we focus on building a domain-specific knowledge graph for research literatures. A similar service is the Semantic Scholar [14] project by Professor Oren Etzioni from Allen Institute for AI. However, Semantic Scholar can only understand a very limited number of relationships (such as 'cite', 'comment', 'use_data_set', and 'has_caption'), and it does not offer entity disambiguation (e.g., mixing all "Wei Wang"'s publication together).

2.1 Knowledge Graphs

Knowledge Graphs contain a valuable of information in a structured format, traditionally originally mined from table-like structures form places like Wikipedia [17] tables [9]. It can be used for a diverse range of applications, such as helping other systems reason about quality of harvested facts[15], provide table-like facts about an entity[6], and question-answering systems[7]. Moreover, recent years have witnessed a surge in large scale knowledge graphs, such as DBpedia [9], Freebase [4], Googles Knowledge Graph [6], and YAGO [15].

The Knowledge Graph name follows from the data structure that is created from the facts in its final form, a graph with nodes representing entities and edges representing various relations between entities. In Figure 2.1, it is possible to observe an example plotted in this form. The list of possible entities classes, and allowable relations between entities is known as a schema. The schema represented in Figure 2.1 is detailed in Table 2.2; one can observe that, as an example, *Max Planck* is an entity of the type *physicist*.

Moreover, based on the facts presented, entailments can be made and one trivial example is denoted in Table 2.1. More complex examples of possible reasoning can be seen in [16]. This is equivalent to traversing the graph

```
type(A, D) :- type(A, B), subclassOf(B, C), subclassOf(C, D)
```

Table 2.1: This example allows one to assert that `type(Max Planck, person)` is also valid based on the fact tuples presented in Table 2.2.

from a node that represents a more specific information, to a node that represents a more general information - e.g.: another possible child node of *scientist* could be the type *biologist*.

```
type(Max Planck, physicist)
subclassOf(physicist, scientist)
subclassOf(scientist, person)
bornIn(Max Planck, Kiel, 1858)
type(Kiel, city)
locatedIn(Kiel, Germany)
hasWon(Max Planck, Nobel Prize, 1919)
```

Table 2.2: Some facts regarding Max Planck, also depicted in Figure 2.1.

This example denotes a classical domain, more precisely important persons, companies, locations, and the relations between them, in which Information Extraction (IE) tools have been very successful on.

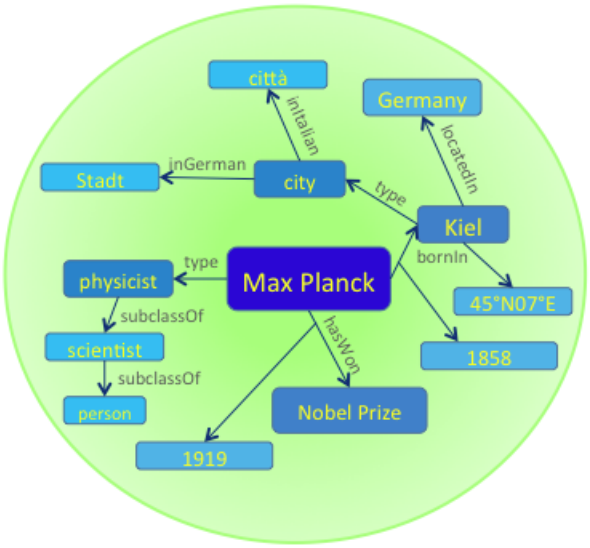

Figure 2.1: An example of knowledge graph from [15] plotted with vertices and edges.

As mentioned previously, YAGO [15] is a prominent Knowledge Graph database, and possesses several advanced characteristics. Every relation in

its database is annotated with its confidence value. See the example of the resulting graph in Figure 2.2. Moreover, YAGO combines the provided taxonomy with WordNet [11] and with the Wikipedia category system [17], assigning the entities to more than 350,000 classes. This allow for very powerful querying. Finally, it attaches a temporal and a spacial dimension to many of its facts and entities, being then capable to answer questions such as *when* and *where* such event took place.

WordNet is a semantically-oriented dictionary of English, similar to a traditional thesaurus but with a richer structure [3]. More specifically, it provides relations to synonyms, hypernyms and hyponyms, among others.

Figure 2.2: An example of patterns existants in YAGO.

2.2 Information Extraction

Information extraction (IE) is the process of obtaining in an automatic fashion facts and information from unstructured text that can be read by a machine [8]. Also according to [8], the IE process is, in general, divided in the following subtasks: Named Entity Recognition (NER), Coreference Resolution, Entity Disambiguation, Relation Extraction (RE), Event Detection, Temporal Analysis. The main subtasks relevant to this report will be described further in this section.

Once the information is extracted it is then used for tasks such as Template Filling [8], or stored as a Knowledge Graph for downstream logical

reasoning or for further queries.

[PER James Cook] was born on 27 October 1728 in the village of [LOC Marton] in [COUNTY Yorkshire].

Table 2.3: An example of Named Entity Recognition (NER).

Name Entity Recognition is the process of, given a sentence, mark what are the entities that are part of it. Once the entity is detected, it needs to be classified within the classes of the given domain - in the spirit of the previous examples this would be e.g.: *City* or *Person*. A few approaches exist for the problem of NER, mostly related to Pattern Matching or Sequence Classification.

Pattern	**Would yield ENTITY of type**
[PERSON] was born	PERSON
in the village of [LOC]	LOCATION
in [LOC]	LOCATION

Table 2.4: Examples of Named Entity Recognition (NER) patterns, based on the sentence from Table 2.3.

Observe, for an example, the sentence in Table 2.3. Several articles regarding prominent figures, either historical or of our current society, can start with the text *[ENTITY] was born*. One approach might be Pattern Matching, which is to mine the input natural language text while looking for this pattern using Regular Expressions - or a Finite-State Automata [8]. The entities found by this pattern would then also receive the *person* class. Other possible patterns can be found in Table 2.4.

Another way to extract entities from text is to frame the NER problem as a Sequence Classification problem. It requires the training of a classifier in which, given the class of the previous word, and other surrounding features of the current word, will attempt to guess if the current word is an entity, and if it is, also guesses its class.

To achieve this, previously annotated data with existing sentences and its entities is needed. The format in which this annotated data is provided varies, however more commonly the IOB format (Table 2.5) is used in several of the NER tools, including NLTK [3] and the popular Stanford Named Entity Recognizer (NER), part of the Stanford CoreNLP [10]. Stanford CoreNLP provides a set of natural language analysis tools.

The IOB format also helps remove ambiguity in case there are two contiguous entities of same class without any word tagged as *O* in between. In practice, these cases are somewhat rare in several domains, and thus a simplified version without the *B-* and *I-* prefixes are used [16].

Word	**Tag**
James	B-PERSON
Cook	I-PERSON
was	O
born	O
on	O
27	B-DATE
October	I-DATE
1728	I-DATE
in	O
the	O
village	O
of	O
Marton	B-LOC
in	O
Yorkshire	B-LOC
.	O

Table 2.5: Example of IOB-formatted sentence used to train classifiers for the Named Entity Recognition (NER) task, based on the sentence from Table 2.3.

The Stanford Named Entity Recognizer (NER), also known as CRF-Classifier [5], provides a general implementation of (arbitrary order) linear chain Conditional Random Field (CRF) sequence models. A CRF is a conditional sequence model which represents the probability of a hidden state sequence given some observations.

Several useful features can be used during the training of a NER CRF classifier model. In Table 2.6, several examples are presented. The Word Shape feature is an interesting addition from recent research, as it captures the notion that most entities are written in capital letters, or starting with capital letter, or containing numbers in the middle of the word, and other specific shapes.

In addition to the above methods another useful technique is the use of gazetteers. Gazetteers are common for geographical data, where government provided lists of names can contain millions of entries names for all manner of locations along with detailed geographical, geologic and political information [8].

Relation Extraction (RE) is the ability to discern the relationships that exist among the entities detected in a text [8], and is naturally the next challenge after being able to detect entities. It can be done using Pattern Matching, Classifiers, or purely by exploiting linguistic data available form a sentence.

Feature	Description
Word	The current word being classified.
N-grams	A feature from n-grams, i.e., sub-strings of the word.
Previous Class	The class of the immediate previous word.
Previous Word	The previous word.
Disjunctive	Disjunctions of words anywhere in the left or right.
Word Shape	The shape of the word being processed captured using. In general replaces numbers with d, x to lowercase letters, and X to upper-case letters.

Table 2.6: Examples of features used to train the CRFClassifier.

The Pattern Matching technique from NER is then built upon in the Relation Extraction step, and now involves more than one entity, yielding binary relations. This approach is used in tools such as PROSPERA [12] or those mined by PATTY [13]. More specifically, examples of patterns mined by PATTY for the *graduatedFrom* relation are seen in Figure 2.3.

actedIn
created
dealsWith
diedIn
directed
graduatedFrom
happenedIn
hasAcademicAdvisor
hasCapital
hasChild
hasWonPrize
holdsPoliticalPosition
influences
isCitizenOf
isKnownFor
isLeaderOf
isLocatedIn
isMarriedTo
isPoliticianOf
livesIn
participatedIn
playsFor
produced
wasBornIn
worksAt

Pattern	Domain	Range	Confidence	SupportCo-occurrence
graduated [[con]] entered;	person	university	1	4
completed [[prp]] university studies in;	person	organization	1	2
attended before studying law at;	person	organization	1	2
sociology at;	person	university	1	2
speaking [[con]] representing;	person	university	1	2
earned in economics from;	person	organization	1	2
graduated from [[det]] department of;	person	university	1	2
pursued [[det]] degree at;	person	university	1	2
met [[prp]] [[adj]] wife [[det]];	person	organization	1	2
[[det]] member [[det]] governing body of;	person	university	1	2
worked [[con]] received from;	person	university	1	2
[[det]] degree in economics from;	person	university	1	2
entered;	person	organization	0.975	24
obtained [[det]] doctorate at;	person	university	0.974	2
majored at;	person	university	0.966	7
[[det]] graduate student in;	person	organization	0.965	4
received in mathematics from;	scientist	university	0.963	3
graduated [[con]] with honors from;	person	organization	0.947	3
accepted [[det]] chair in;	person	university	0.947	2

Figure 2.3: An example of patterns extracted from PATTY for the *graduatedFrom* relation.

PROSPERA's main technique is that not only it obtain facts based on a small set of initial seed patterns, but also obtain new candidate patterns based on the mined facts. Once the process finishes, these new candidate patterns are evaluated and then added to the the existing pattern repository for re-use. The whole process then iterates again finding even more facts from these new patterns, and new candidate patterns.

- Classification
- Possible features
- Open Information Extraction
- Evaluation
- What drives the field forward

To be completed: entity disambiguation?

2.3 Natural Language Processing

The linguistic data of the text is the base for features in which the classifiers for Information Extraction act on.

To be completed: basic concepts?
To be completed: POS tags?
To be completed: Dependency path?
To be completed: coreference resolution?
To be completed: querying a corpus?
To be completed.

Chapter 3

Challenges with Academic Text

Difficult to manually tag.

Relations that could be extracted, e.g. denes, don't appear with relevant Entities.

A lot of coreference problems, e.g. "their work". Even when their work is simply a paper reference - with unclear. One could trivially parse the above reference with the actual Entity name of the system or algorithm or technique elabrated in the reference paper as to mine the relations between the proper entities.

Methods:

* Use semi-automatic methods to collect research literature and convert them into plain text with certain markups. * Use existing open source solutions to parse the store the input data. * Build a pipeline to extract entities and relationship from the input data. * Perform entity disambiguation and linking to a reference Knowledge Graph (e.g., Yago2 or DBPedia). * Design effective postprocessing methods to improve the quality of the extraction. * Evaluate the entire extraction system.

Chapter 4

Developed Workflow

4.1 Tools

Python

- Java
- BeautifulSoup
- PDF extraction generates noisy output
- NLTK
- Brat
- standoff2conll
- corpkit
- corenlp-xml
- Parsey
- Spacey
- Stanford CoreNLP
- Tregex

4.2 Developed process and new scripts

We leverage on existing pipeline for NER and Relatino Extraction (Stanford) , instead of independent tools.

We use IO notatoion instead of IOB notation notation3 for entity mention la- bels (e.g., the labels for the tokens over the Seattle Seahawks on Sunday (from Figure 1) are encoded as O O NFLTEAM NFLTEAM O DATE). The IO notation facilitates faster inference than the IOB or IOB2 notations with minimal impact on performance, when there are fewer adjacent mentions with the same type. AS PER STANFORD RELATION EXTRACTOR PAPER.

TALK ABOUT FEATURES WE PICKED FOR NER.

TALK ABOUT FEATURES WE PICKED FOR REL.

Open information extraction (open IE) has been shown to be useful in a number of NLP tasks, such as question answering (Fader et al., 2014), rela- tion extraction (Soderland et al., 2010), and infor- mation retrieval (Etzioni, 2011).

We did not implement coreference.

Expected Outcome:

1. A system that can build a knowledge graph from research literatures. 2. A written report about the detailed designs and implementations of this system. 3. A seminar to present the process and outcome of this project.

Chapter 5

Results

Chapter 6

Conclusion

Furhter ideas: - Research relations through time. You could have a certain feature. - Reinforcements made to definitions in other papers. - Events, such as changes in conclusions - e.g.: this was the better technique, now this other technique is the best.

Bibliography

[1] *Adobe: What is PDF?* https://acrobat.adobe.com/us/en/why-adobe/about-adobe-pdf.html. [Online; accessed 14-August-2016]. 2016.

[2] *Bing Knowledge and Action Graph.* https://www.bing.com/partners/knowledgegraph. [Online; accessed 14-August-2016]. 2016.

[3] Steven Bird, Ewan Klein and Edward Loper. *Natural Language Processing with Python.* O'Reilly Media, 2009.

[4] Kurt Bollacker et al. 'Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge'. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data.* SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 1247–1250. ISBN: 978-1-60558-102-6.

[5] Jenny Rose Finkel, Trond Grenager and Christopher Manning. 'Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling'. In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics.* ACL '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 363–370.

[6] *Google Knowledge Graph.* https://www.google.com/insidesearch/features/search/knowledge.html. [Online; accessed 14-August-2016]. 2016.

[7] Ben Hixon, Peter Clark and Hannaneh Hajishirzi. 'Learning Knowledge Graphs for Question Answering through Conversational Dialog'. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Denver, Colorado: Association for Computational Linguistics, 2015, pp. 851–861.

[8] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. ISBN: 0130950696.

[9] Jens Lehmann et al. 'DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia'. In: *Semantic Web Journal* 6.2 (2015), pp. 167–195.

[10] Christopher D. Manning et al. 'The Stanford CoreNLP Natural Language Processing Toolkit'. In: *Association for Computational Linguistics (ACL) System Demonstrations*. 2014, pp. 55–60.

[11] George A. Miller. 'WordNet: A Lexical Database for English'. In: *Commun. ACM* 38.11 (Nov. 1995), pp. 39–41. ISSN: 0001-0782.

[12] Ndapandula Nakashole, Martin Theobald and Gerhard Weikum. 'Scalable Knowledge Harvesting with High Precision and High Recall'. In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. WSDM '11. New York, NY, USA: ACM, 2011, pp. 227–236.

[13] Ndapandula Nakashole, Gerhard Weikum and Fabian Suchanek. 'PATTY: A Taxonomy of Relational Patterns with Semantic Types'. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 1135–1145.

[14] *Semantic Scholar*. http://allenai.org/semantic-scholar/. [Online; accessed 20-August-2016]. 2016.

[15] Fabian M. Suchanek, Gjergji Kasneci and Gerhard Weikum. 'Yago: A Core of Semantic Knowledge'. In: *Proceedings of the 16th International Conference on World Wide Web*. WWW '07. New York, NY, USA: ACM, 2007, pp. 697–706. ISBN: 978-1-59593-654-7.

[16] Mihai Surdeanu et al. 'Customizing an Information Extraction System to a New Domain'. In: *Proceedings of the ACL 2011 Workshop on Relational Models of Semantics*. RELMS '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 2–10.

[17] *Wikipedia, The Free Encyclopedia*. http://www.wikipedia.org/. [Online; accessed 21-August-2016]. 2010.