Extraction of Career Profiles from Wikipedia

Firas Dib, Simon Lindberg, Pierre Nugues

Lund University
LTH, Department of Computer Science, S-221 00 Lund, Sweden
ada10fdi@student.lu.se, ada10sli@student.lu.se, Pierre.Nugues@cs.lth.se

Abstract

In this paper, we describe a system that gathers the work experience of a person from her or his Wikipedia page. We first extract an ontology of profession names from the Wikidata graph. We then parse the Wikipedia pages using a dependency parser and we connect persons to professions through the analysis of parts of speech and dependency relations we extract from text. Setting aside the dates, we computed recall and precision scores on a very limited and preliminary test set for which we could reach a recall of 74% and a precision of 95%, showing our approach is promising.

Keywords: Knowledge extraction, Wikidata ontology, Dependency parsing

1. Introduction

Biographies form a category of their own in literature as they typically mix free-form text – a life narrative – with a set of well-defined numerical and nominal properties, such as dates of birth and death, country of origin, titles and decorations, etc. that merely resort to databases. Textual biographies can therefore be associated to structured databases that describe such properties in the form of tables or graphs. Texts and databases are both useful and complementary for humanities research. While text often contains more details on people's life, databases enable researchers to formulate questions like:

> Are there welders who became prime ministers?

and immediately have answers.

Although there are now scores of digital biographies, Wikipedia has become the major reference of the internet. It is free and easy to download; it covers more people than other online resources; it is open to popular culture; and it is multilingual. This makes it unique even if Wikipedia often reuses text and data from older printed biographies and contains mistakes. In addition to its scope and size, there are many open computer tools that have been designed for Wikipedia that make the development of new programs dedicated to this resource faster.

We created a system that takes a corpus of Wikipedia pages describing people as input and that outputs a career profile for each respective individual. To carry this out, we used available tools to parse and extract information from text and then analyze the data.

A practical use of our system could be to expand Wikidata, the data repository companion to Wikipedia. As of today, Wikidata often associates people with the most notable occupation of their life. The system we describe makes it possible to build more comprehensive semantic knowledge bases of career timelines as it extracts all the occupations, possibly secondary, mentioned in the text.

In our experiments, we used the Swedish version of Wikipedia and we are strictly dependent on the Wikipedia format. Nonetheless, we only used the text itself so this source could easily be replaced with another one in another language and another format. In addition, beyond Wikidata, the techniques we have developed could be applied to expand any database.

2. Previous Work

The analysis of career profiles from biographies is a specific case of information extraction that produces tabular data from raw text. Information extraction has a long history in natural language processing, starting from the message understanding conferences (MUC) (Grishman and Sundheim, 1996), and has been carried out with a variety of techniques along the time: rule-based, statistical, or hybrid, with a current focus on machine learning (Mausam et al., 2012). See Hobbs et al. (1997) for the description of an early and oft-cited system and Roche and Schabes (1997) for a review.

There are a few papers describing the extraction of timelines from Wikipedia. Timely YAGO (Wang et al., 2010) is an example of them that is limited to the analysis of infoboxes, summaries of facts in the form of tabular data inside the articles, and lists in articles. Exner and Nugues (2011) is another example that uses semantic role labeling and the LODE model (Shaw, 2010) to extract events. Wu and Weld (2010) is a third example that combines Wikipedia infoboxes and document text to collect data to train relation classifiers.

Contrary to these works, the system we describe is dedicated to the extraction of careers through the analysis of the dependency graphs of the sentences. To collect the vocabulary associated with occupations, the system creates a career ontology that it automatically retrieves from the Wikidata repository. In addition to being automatic, this process can easily be extended to create multilingual vocabularies.

3. Term Extraction

3.1. Wikidata: A Semantic Repository

We used Wikidata as main source of structured knowledge on human beings and their occupations. Wikidata is a free data repository from the wikimedia foundation. Wikidata started as a means to identify named entities across all their Wikipedia language versions with a unique number.

Göran Persson		Jacques Delors	
an	Göran Persson	bg	Жак Делор
be	Ханс Ёран Персан	ca	Jacques Delors
bg	Йоран Пешон	cs	Jacques Delors
bn	জোরান পের্শোন	da	Jacques Delors
ca	Göran Persson	de	Jacques Delors
cs	Göran Persson	el	Ζακ Ντελόρ
da	Göran Persson	en	Jacques Delors
de	Göran Persson	eo	Jacques Delors
en	Göran Persson	es	Jacques Delors
eo	Göran Persson	et	Jacques Delors

Figure 1: Links from Wikidata to articles on Göran Persson and Jacques Delors in ten different languages

Göran Persson, for instance, a former Prime Minister of Sweden, has the identifier Q53747 in Wikidata that links this entity to the 44 different language versions of his biography in Wikipedia, while Jacques Delors, a former president of the European Commission, has the identifier Q153425 that provides links to the 35 language versions of Delors' biography. Figure 1 shows the 10 first links for these two persons with their language codes, for instance *en* for English, *de* for German, or *el* for Greek and their name's transcription in the corresponding script as in Greek: Ζακ Ντελόρ, for Jacques Delors.

The entities reflected by Q-numbers are linked to concepts or other entities by a set of properties that describes the entity, P_x, where x is a number. Property P31, corresponding to **instance of**, applies to Göran Persson with the value *human*; P569, date of birth, with the value *20 January 1949*; P26, spouse, *Anitra Steen*; P106, occupation, *politician*, etc.

```
P31(Q53747) = human
P569(Q53747) = 20 January 1949
P26(Q53747) = Anitra Steen
P106(Q53747) = politician
```

The values *human*, *Anitra Steen*, and *politician* having themselves unique Q-numbers, respectively Q5, Q444325, and Q82955.

The P39 property, **position held**, tracks the career of a person and consists of multiple values. Wikidata lists five positions held by Göran Persson: Leader of the Opposition, Minister for Finance, Skolminister (Minister for Schools), Prime Minister of Sweden, and Member of the Riksdag, possibly with time values or boundaries (Fig. 2).

Wikidata stores all this information as a graph in the RDF format. It is similar to earlier projects such as DBpedia (Auer et al., 2007), Yago (Suchanek et al., 2007), or Freebase (Bollacker et al., 2008). A key difference between these earlier works and Wikidata is that Wikidata is language-agnostic and an integral part of the Wikipedia structure.

3.2. Extracting Occupations

The properties such as P106, *occupation*, are organized as hierarchies of more specific properties. In the case of *occupation*, Figure 3 shows an excerpt of such a hierarchy, where Wikidata gathers all types of jobs, professions, and careers.

Figure 2: The two first positions of Göran Persson out of five, where Minister of Finance has a start date, 7 October 1994, and an end date, 22 March 1996

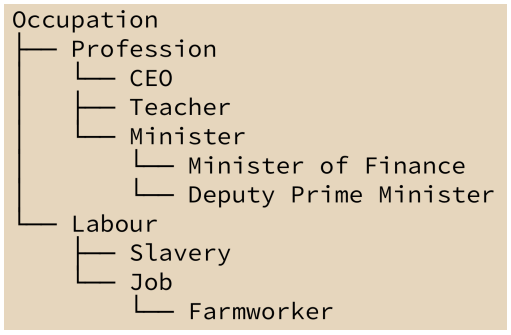


Figure 3: An excerpt of the Wikidata ontology starting from the occupation node

We processed the Wikidata graph and the concept hierarchies to create a baseline list of professions. We considered the **Instance of** (P31) and **Subclass of** (P279) properties that we took as guiding relations to extract the people careers. We created a list of terms using all the descendants of the `Occupation` node that we chose as the root node since `Profession`, `Job`, and `Labour` are all an **Instance of** `Occupation`. We created this list in a preprocessing stage, independently and prior to the actual article parsing.

4. NLP Pipeline

Although Wikidata covers lots of biographical details, it is far from being exhaustive and much of the information on the career timelines still lays in the text. Stefan Löfven, another Prime Minister of Sweden, provides an example of this, where his Wikipedia page in English states that:

> Löfvén began his career in 1979 as a welder at Hägglunds in Örnsköldsvik.

while Wikidata only lists him as a politician[1]. We assembled a pipeline of natural language processing components to analyze the text and extract such information.

[1]Both the Wikidata item and the Wikipedia page were retrieved on May 28, 2015.

We downloaded the Swedish version of Wikipedia and we first processed the articles to remove the wiki markup. This markup code enriches the text of Wikipedia articles, for instance to create the links or to identify the section titles. We then applied a part-of-speech tagger and a dependency parser to the text. We split the Wikipedia archive in chunks allowing for a multithreaded execution in order to speed up the process:

1. The first step of the pipeline was to parse and remove the Wikipedia markup. This markup is functionally similar to HTML or XML, but has a different format that requires a different parser. We used the Sweble tool (Dohrn and Riehle, 2013) to carry it out.

2. We then applied a tagger to the text in Swedish and annotate the words with their parts of speech. We used Stagger (Östling, 2013) that also includes a named entity recognition (NER). We used these named entities further down in the pipeline to extract the persons from the sentence.

3. Finally, we ran the Maltparser dependency parser (Nivre et al., 2006) on the POS tagged sentences to have a syntactic representation of them.

Table 1 shows the pipeline, its components, and for each component its input and output.

Input	Tool	Output
Wikipedia article	Sweble	Plain text
Plain text	Stagger	POS tagged
POS tagged	Maltparser	Dependency parsed
Dep. parsed	Career profiler	Timeline

Table 1: NLP processing pipeline

5. Career Parsing

The career parsing module analyzes the text, sentence by sentence, to find out the persons, what they are working at, during what time frame, and tries to connect these elements together through the dependency graph of the sentence.

5.1. Finding Persons

The first step of the career parser identifies the mentions of human beings in each sentence. We applied the following rules to decide if a word referred to a person:

1. The word matches a regular expression based on the Wikipedia page title: The person the page is about;

2. The word is a singular pronoun in Swedish: *han* "he", *hon* "she", *hans* "his", or *hennes* "her";

3. The word is tagged as a person by Stagger's named entity recognizer.

We stored all the persons we found as well as the sentences they occurred in.

5.2. Finding Jobs

The second step finds the job names mentioned in the sentences. We used the list of professions we collected from Wikidata in Sect. 3.2. to check the presence of corresponding words and extract them. However, this initial profession list is far from being exhaustive and we applied additional rules to complete it. To decide if a given a word in a sentence was a profession, we checked if it was:

1. A job name in the list, without any modification;

2. The compounding of two stems, where the last one is a profession in the list. We split the word in a prefix and a suffix and we applied a greedy search on the suffix, where both the prefix and suffix had to be in a dictionary of Swedish words. The prefix check was done to eliminate false positives such as *kre**tsar*** "circuits" that could be interpreted as *tsar* "Czar" preceded by a meaningless prefix *kre*.

3. The compounding of two stems separated by a linking morpheme (fogemorpheme). In Swedish, and other Germanic languages, it is common to either add an **s** between the two stems or change the last vowel of the first stem. We used two simple morphology rules to extract them:

   (a) If the last letter of the prefix ends with an **s**, we remove it and we check if this prefix is a valid word in the dictionary as with *utbildning**s**minister*, (utbildning + minister), "Minister for Schools".

   (b) If the last letter of the prefix ends in a vowel, we replace it by another vowel and we see if this prefix makes up a word as with: *förskol**e**lärare* (förskola + lärare) "preschool teacher".

5.3. Finding Verbs

As noted by Tesnière (1966), verbs in European languages are central elements to describe processes between actors and circumstances. We started from this observation and we extracted the verbs hinting at a professional activity from the sentences. As vocabulary, we used the following set of Swedish verbs: *vara* "be", *bli* "become", *arbeta* "work", *jobba* "work", and *praktisera* "practice".
We then considered that these verbs were potential linking nodes to relate a person to a job in sentence.

5.4. Finding a Path

The path finding step links people to jobs. From the previous steps, the career parser has gathered for each sentence respective lists of persons, jobs, and verbs. We create a path between these words by traversing the dependency graph of a sentence until we find a common ancestor.
Figure 4 shows the dependency graph of the sentence:

> Hon var tidigare kommun- och regionminister 2001-2005.
> "Previously, she served as minister for municipalities and regions (2001-2005)",

where we link a person mentioned by the feminine singular pronoun *hon*, highlighted in green in the figure to a profession, *regionminister*, in turquoise, through the verb *var*, in purple, and where we extract the path:

$$\text{hon} \rightarrow \text{var} \leftarrow \text{och} \leftarrow \text{regionsminister}$$

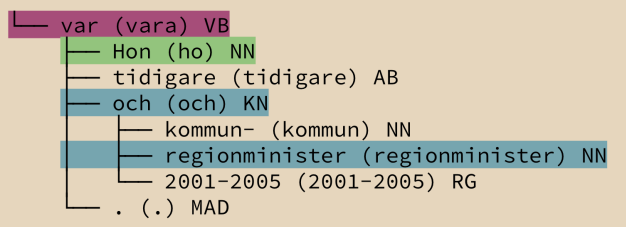


Figure 4: Dependency tree of the sentence *Hon var tidigare kommun- och regionminister 2001-2005* and the path between a person and a job

Figure 5 shows another example with the sentence:

> Hans Göran Persson är en svensk politiker som var statsminister 1996-2006.
> "Hans Göran Persson is a Swedish politician that was Prime Minister between 1996 and 2006."

where the career parser connects a person, *Göran Persson*, to two occupations, *politiker* "politician" and *statsminister* "Prime Minister", through the verbs *är* "is" and *var* "was".
To deal with the case where multiple persons are referenced in a sentence alongside a job, we introduced two additional constraints:

1. The path from job to person must include one of the professional activity verbs;
2. This path must be the shortest one. We search all the paths between all the persons and all the jobs and we keep the shortest path for each respective profession. Figure 6 shows an example of this in a sentence with two persons and one activity.

5.5. Finding Dates

Once we have linked an occupation to a person, we extract the dates from the sentence. We implemented a simple procedure, where we looked at the words preceding and following the word representing the job.
We first try to match the adjacent words to a date expression. If these words correspond to dates, we use them to annotate the occupation with time stamps; if the adjacent words are prepositions or conjunctions, we skip them and we repeat the matching attempt.

6. Results

We processed the complete collection of Swedish Wikipedia articles referring to a person in Wikidata. We extracted a total of 267,786 jobs from 170,300 articles. Figure 7 shows the seven professions we obtained for *Barack Obama*:

- *President*,
- *Ämbete* "officer",
- *Senator*,
- *Handledare* "instructor, supervisor",
- *Konsult* "consultant",
- *Sommaranställd* "Summer employee", and
- *Journalist*,

while Figure 8 shows the three ones for *Filippa Reinfeldt*:

- *Politiker*, "politician",
- *Talesperson*, "spokesperson", and
- *Sjöofficer*, "naval officer".

The third profession of *Filippa Reinfeldt* is wrong and corresponds to that of her father.
We assessed the accuracy of the system using a small and preliminary test set of 10 random Wikipedia articles about people that were about one or two paragraphs long (Table 2). Since the articles were short, they were often to the point and did not contain any complicated language. This made the recall easier than if we would have tested against larger and more complex articles.
Although a more thorough testing would be necessary to validate the system, it shows the promising nature of our approach.

Recall	Precision	F-score
74.1%	95.2%	83.3%

Table 2: Recall and precision

7. Further Work

The techniques we described in this paper could be improved in many ways. Here is a list of possible further work:

Negations. We did not consider negations, such as *inte* "not" or *aldrig* "never", in the sentences. This is an aspect that could be improved;

Activities. We only extracted actual occupations and we did not associate work related activities or references to a workplace with a profession, meaning that neither phrase *writes articles* nor *works at the New York Times* would relate the person to a journalist or writer occupation.

Wikidata limitations. The search performed to find all the occupations collects anything remotely related to the **Occupation** node. This results in an overgeneration. A more robust analysis would filter out erroneous professions, for example, by controlling that they do not have a path to *business* or *superheroes*.

Naive name matching. The procedure we used is naive and a named entity linker would certainly improve the results.

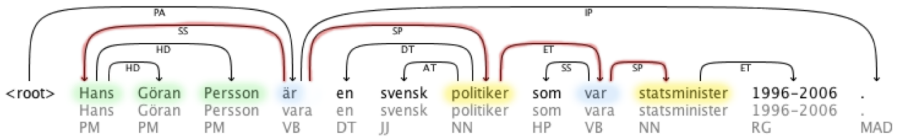


Figure 5: Dependency tree and the path between a person and a job

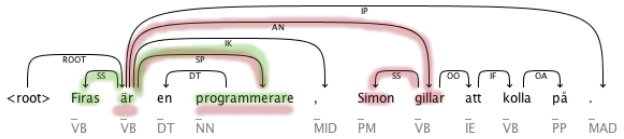


Figure 6: Two competing paths between *Firas* and *programmerare* and *Simon* and *programmerare*, where the selected one is the shortest

Coreference. While looking for persons in the sentence, we also check for pronouns. We then assume that the pronouns are referring to the person of interest. A coreference solver would make this step more accurate.

Swedish only. Our system only supports the Swedish language. It would however be relatively simple to adapt it to English as well.

8. Acknowledgements

This research was supported by Vetenskapsrådet and the *Det digitaliserade samhället* program.

9. References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *The Semantic Web, Proceedings of 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007*, pages 722–735, Busan, Korea, November 11-15. Springer.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. 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, pages 1247–1250.

Hannes Dohrn and Dirk Riehle. 2013. Design and implementation of wiki content transformations and refactorings. In *Proceedings of the 9th International Symposium on Open Collaboration*, WikiSym '13, pages 2:1–2:10.

Peter Exner and Pierre Nugues. 2011. Using semantic role labeling to extract events from Wikipedia. In *Proceedings of the Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011). Workshop in conjunction with the 10th International Semantic Web Conference 2011 (ISWC 2011)*, Bonn, October 23–24.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference – 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, volume 1, page 466–471, Copenhagen.

Jerry R. Hobbs, Douglas E. Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1997. FASTUS: a cascaded finite-state transducer for extracting information from natural-language text. In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing*, chapter 13, pages 383–406. MIT Press, Cambridge, Massachusetts.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 523–534.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC2006)*.

Robert Östling. 2013. Stagger: an open-source part of speech tagger for Swedish. *Northern European Journal of Language Technology*, 3.

Emmanuel Roche and Yves Schabes, editors. 1997. *Finite-State Language Processing*. MIT Press, Cambridge, Massachusetts.

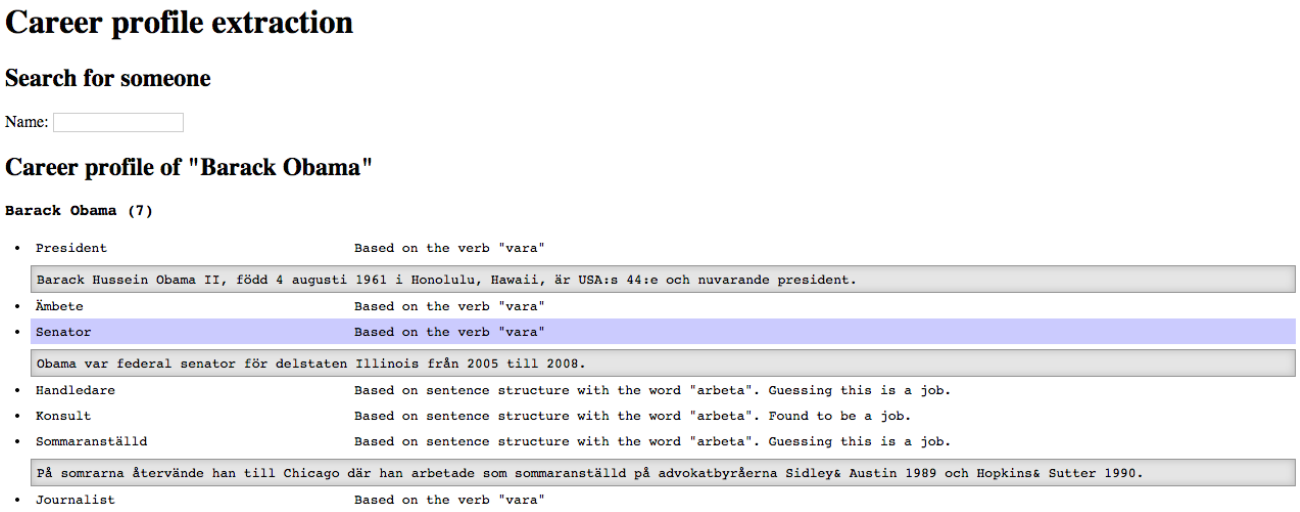
Ryan Benjamin Shaw. 2010. *Events and Periods as Concepts for Organizing Historical Knowledge*. Ph.D. thesis, University of California, Berkeley.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, Banff. ACM.

Lucien Tesnière. 1966. *Éléments de syntaxe structurale*. Klincksieck, Paris, 2nd edition.

Yafang Wang, Mingjie Zhu, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. 2010. Timely YAGO: Harvesting, querying, and visualizing temporal knowledge from wikipedia. In *Proceedings of the 13th International Conference on Extending Database Technology*, EDBT '10, pages 697–700.

Fei Wu and Daniel S. Weld. 2010. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 118–127.



Career profile extraction

Search for someone

Name:

Career profile of "Barack Obama"

Barack Obama (7)

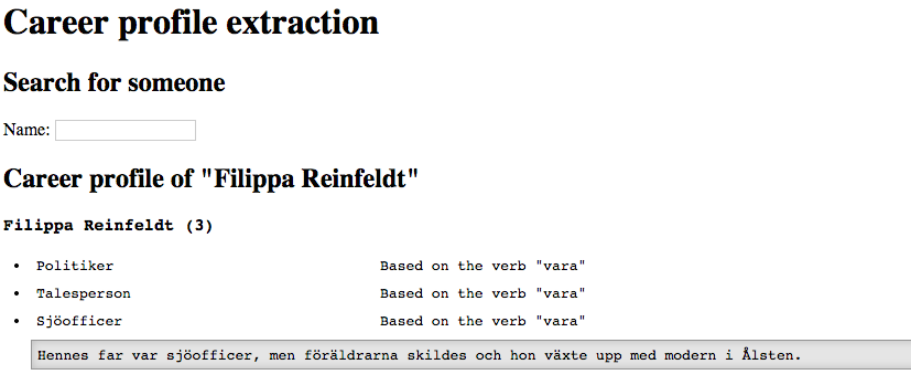
- President — Based on the verb "vara"

  Barack Hussein Obama II, född 4 augusti 1961 i Honolulu, Hawaii, är USA:s 44:e och nuvarande president.
- Ämbete — Based on the verb "vara"
- Senator — Based on the verb "vara"

  Obama var federal senator för delstaten Illinois från 2005 till 2008.
- Handledare — Based on sentence structure with the word "arbeta". Guessing this is a job.
- Konsult — Based on sentence structure with the word "arbeta". Found to be a job.
- Sommaranställd — Based on sentence structure with the word "arbeta". Guessing this is a job.

  På somrarna återvände han till Chicago där han arbetade som sommaranställd på advokatbyråerna Sidley& Austin 1989 och Hopkins& Sutter 1990.
- Journalist — Based on the verb "vara"

Figure 7: Timeline extracted from the article on Barack Obama



Career profile extraction

Search for someone

Name:

Career profile of "Filippa Reinfeldt"

Filippa Reinfeldt (3)

- Politiker — Based on the verb "vara"
- Talesperson — Based on the verb "vara"
- Sjöofficer — Based on the verb "vara"

  Hennes far var sjöofficer, men föräldrarna skildes och hon växte upp med modern i Ålsten.

Figure 8: Timeline extracted from the article on Filippa Reinfeldt