Constructing a Knowledge Base on Aging

An Automated Approach

Mark Farrell

Bioinformatics Researcher

Center for Research and Education on Aging
Lawrence Berkeley National Laboratory
University of California, Berkeley

September 4th, 2014

Outline

1. Automatically Constructing Knowledge Bases

2. Extracting Facts in a Structured Format

3. Results & Discussion

Overview

- CREA is constructing a knowledge base to study and understand the human aging process.

Overview

- CREA is constructing a knowledge base to study and understand the human aging process.
- New discoveries are published quickly and in large volume.

Overview

- CREA is constructing a knowledge base to study and understand the human aging process.
- New discoveries are published quickly and in large volume.
- It is infeasible to construct the knowledge base by hand.

Overview

- CREA is constructing a knowledge base to study and understand the human aging process.
- New discoveries are published quickly and in large volume.
- It is infeasible to construct the knowledge base by hand.
- Working on software to construct the knowledge base automatically.

Introduction

How to Automatically Construct the Knowledge Base

- Routinely search for keywords related to aging, dowloading text articles from sources like PubMed and WebMD.

Introduction

How to Automatically Construct the Knowledge Base

- Routinely search for keywords related to aging, dowloading text articles from sources like PubMed and WebMD.
- Build a spam filter to get rid of non-scientific sentences.

Introduction

How to Automatically Construct the Knowledge Base

- Routinely search for keywords related to aging, dowloading text articles from sources like PubMed and WebMD.
- Build a spam filter to get rid of non-scientific sentences.
- Extract scientific facts from the sentences and save them in a structured format.

Introduction

How to Automatically Construct the Knowledge Base

- Routinely search for keywords related to aging, dowloading text articles from sources like PubMed and WebMD.
- Build a spam filter to get rid of non-scientific sentences.
- Extract scientific facts from the sentences and save them in a structured format.
- Provide a graphical interface that allows users to search and otherwise explore the knowledge base.

Summary of Progress

- Devised and implemented the method for finding simple facts in sentences, extracting them in a structured format.

Summary of Progress

- Devised and implemented the method for finding simple facts in sentences, extracting them in a structured format.
- Began work on a web viewer for the knowledge base.

Outline

1 Automatically Constructing Knowledge Bases

2 Extracting Facts in a Structured Format

3 Results & Discussion

Tokenization

- Input a text document and read it, one sentence at a time.

Example: Tokenization

```
scala> tokens(”The man walks. The dog eats.”)
res0: List[String] = List(The man walks., The dog eats.)
```

Parsing

- For each sentence, generate a constituent tree that describes its phrase structure.

Example: Parsing

```
scala> parse(”The man walks the dog.”)
res0: Tree[String] = (ROOT
  (S
    (@S
      (NP (DT The) (NN man))
      (VP (VBZ walks)
        (NP (DT the) (NN dog))))
    (. .)))
```

Parsing Method

- The University of Pennsylvania Treebank Project:

Parsing Method

- The University of Pennsylvania Treebank Project:
    - Defines notation for constituent trees.

Parsing Method

- The University of Pennsylvania Treebank Project:
  - Defines notation for constituent trees.
  - Parses sentences from the Wall Street Journal by hand.

Parsing Method

- The University of Pennsylvania Treebank Project:
  - Defines notation for constituent trees.
  - Parses sentences from the Wall Street Journal by hand.
- The Berkeley Parser is software that guesses how to parse a sentence from the notation and examples specified by the Penn Treebank.

Compilation

- Extract facts from each constituent tree.

Example: Compilation

```
scala> compile(”The man walks the dog.”).shows
res0: String = [<compound:walk(<atom:man>, <atom:dog>)>]
```

Compilation Method

Pattern match on the constituent trees. Define patterns for:

1. Extracting nouns from noun phrases (NP).

Compilation Method

Pattern match on the constituent trees. Define patterns for:

1. Extracting nouns from noun phrases (NP).
2. Extracting predicates and nouns from verb phrases (VP).

Compilation Method

Pattern match on the constituent trees. Define patterns for:

1. Extracting nouns from noun phrases (NP).
2. Extracting predicates and nouns from verb phrases (VP).
3. Extracting facts from complete clauses (S), making logical assertions with nouns and predicates.

Outline

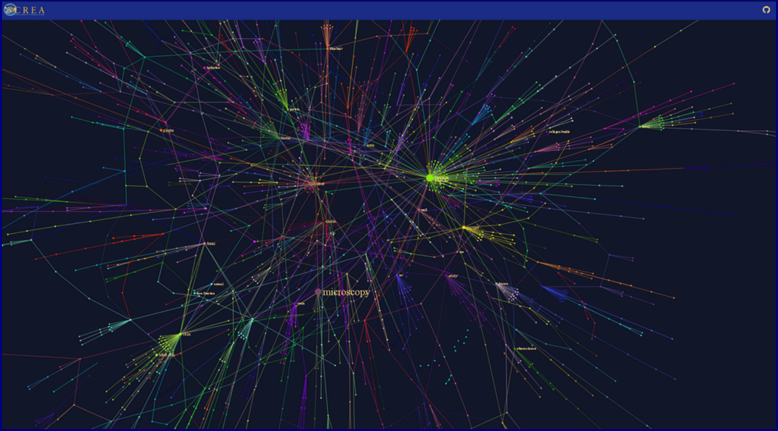
1 Automatically Constructing Knowledge Bases

2 Extracting Facts in a Structured Format

3 Results & Discussion

Software Demonstration

A preview of CREA's knowledge base, compiled from PubMed abstracts.



Performance

Parallelization

- It is possible to extract facts from many sentences at the same time.

Accuracy

- Filter spam sentences from documents.

Accuracy

- Filter spam sentences from documents.
- The accuracy of the parser could be optimized:

Accuracy

- Filter spam sentences from documents.
- The accuracy of the parser could be optimized:
  - Should be trained to identify more nouns from the biomedical domain.

Accuracy

- Filter spam sentences from documents.
- The accuracy of the parser could be optimized:
    - Should be trained to identify more nouns from the biomedical domain.
- Define more patterns for extracting facts:

Accuracy

- Filter spam sentences from documents.
- The accuracy of the parser could be optimized:
  - Should be trained to identify more nouns from the biomedical domain.
- Define more patterns for extracting facts:
  - The software succeeds around 50% of the time.

Missing Features

- Support negated clauses and conditional logic.

Missing Features

- Support negated clauses and conditional logic.
- Facts can contradict each other:

Missing Features

- Support negated clauses and conditional logic.
- Facts can contradict each other:
    - Store the probability that is true as the weight of its edge on the knowledge base's graph.

Missing Features

- Support negated clauses and conditional logic.
- Facts can contradict each other:
  - Store the probability that is true as the weight of its edge on the knowledge base's graph.
- Scale and launch the software service.

Conclusion

- Demonstrated a method for automatically constructing CREA's knowledge base on aging.

Conclusion

- Demonstrated a method for automatically constructing CREA's knowledge base on aging.
- Showed how to extract facts from English text in the knowledge base's structured format.

Conclusion

- Demonstrated a method for automatically constructing CREA's knowledge base on aging.
- Showed how to extract facts from English text in the knowledge base's structured format.
- Discussed the need to improve software accuracy by lensing in on the biomedical domain.

Conclusion

- Demonstrated a method for automatically constructing CREA's knowledge base on aging.
- Showed how to extract facts from English text in the knowledge base's structured format.
- Discussed the need to improve software accuracy by lensing in on the biomedical domain.
- Suggested how the software implementation can be scaled for production usage.