Vybrané aktuální projekty Centra ZPJ

Vojtěch Kovář, Zuzana Nevěřilová

E-mail: xkovar3@fi.muni.cz,xpopelk@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

- Automatické rozpoznání významů slov
- TextPerience: Experience the Text

Automatické rozpoznání významů slov

Kontext:

- ,,One-click dictionary”
- post-editing lexicography

Cíl – word sense iduction, WSI:

- vstup: označkovaný korpus + slovo
- výstup: rozdělení významů pro dané slovo
- výstup: indikátory významů

Usnadnění práce pro editora slovníku:

- pouze editace draftu hesla
- DWS (dictionary writing system) musí podporovat snadnou editaci významů

Metody

Kolokace a dokumenty:

- dokument obvykle obsahuje pouze 1 význam
- jedna kolokace často znamená 1 význam

Word embeddings:

- vektor slova může být součtem více vektorů
- = významů
- hledáme nejlepší rozdělení

Kolokace a dokumenty

Algoritmus:

- pokud má kolokace dost dokumentových souvýskytů s nějakou kolokací v již existujícím clusteru, přidej ji do clusteru
- jinak vytvoř nový cluster
- dost = funkce frekvence slova (1/5000)
- významy = clustery
- indikátory významů = kolokace

Problémy:

- princip "1 sense per collocation" ne vždy funguje
- kolokace se všemi významy slova
- parametry jsou důležité
- (min. freq. = 100, min. score = 3)

Příklad

Word embeddings

Skip-gram:

- pro každé slovo W
- pravděpodobnost, že určité slovo se vyskytne v okolí W

Adaptive skip-gram (Adagram):

- pro každé slovo W a index významu
- pravděpodobnost, že určité slovo se vyskytne v okolí W
- musíme předem určit počet významů

Výstup:

- vektory pro jednotlivé významy
- indikátory významů = podobná slova (blízké vektory)

Adagram – příklad

Lodička:

- střevíček kozačka botka bota páskový balerína balerínka podpatek
- člun kajak pramice kanoe parník loďka loďka kánoe
- skořápka skořápkový rozkrajování loďka košíček pouštění zvoneček věneček
- loďka sandál bota botka teniska vratký polobotka bosý
- teniska tričko mikina gumáček vestička gumák bota džíny

Úspěšnost

Základní vyhodnocení (10 víceznačných slov, asi 30 významů):

	Precision	Recall
Docs CZ	46 %	61 %
Adagrams CZ	38 %	61 %
Docs DK	67 %	84 %
Adagrams DK	18 %	35 %

Poznámky:

- čistota korpusu může mít velký vliv
- špatně definovaný úkol, spolehlivé vyhodnocení je problém

Výhody a nevýhody

Kolokace a dokumenty:

- závislé na gramatice pro slovní profily (sketch grammar), značkování apod.
- problém s kolokacemi všech významů
- technicky přímočaré
- propojené s korpusovým manažerem
- snadno se ladí

Word embeddings:

- černá skříňka – prakticky nejde ladit
- musíme předem určit počet významů
- není snadné určit, do kterého významu patři konkrétní výskyt
- výpočetně náročné
- nezávislé, potřebujeme pouze texty

Možná vylepšení

Kombinace:

- vytvoříme vektory pro kolokace, ty pak budeme clusterovat

Detekce kolokací se všemi významy:

- vytvoříme matici dokumentových souvýskytů všech kolokací
- hledáme skupinky
- kolokace se všemi významy bude možné poznat

TextPerience: Experience the Text

Upload a document or enter an URL

File uploaded
120932 bytes
HyperText Markup Language
(HTML)

Extract text from the document

The text extraction quality depends on the data format and the way it was created.
Extraction method Justext, the content is encoded in utf-8.
language: English

SEE THE TEXT

See what we can recognize in the text

We use linguistic analysis, corpus tools, and mathematics.

TextPerience: Experience the Text

- a demo application
- complete processing pipeline from data to knowledge:
  1. convert an uploaded file to text
  2. detect language
  3. apply keyword search
  4. recognize named entities
  5. perform Wikipedia search of the keywords
  6. evaluate the results
- so far, keywords in 12 languages (cs, sk, en, de, fr, it, es, vi, pl, hu, da, ru)
- so far, named entities in 5 languages (cs, en, es, de, ru)

TextPerience I: Extract Texts from Files

After a successful MIME type guessing ...

- use Apache Tika for Office-like texts and PDFs
- use JusText for extracting text from web pages, Pomikálek (2011)
- take plain texts as they are

many other file types:

- tables (Excel, TSV, CSV)
- presentations
- images (OCR)
- source codes (plain text with a particular extension)
- videos (subtitles)
- ...

TextPerience I: Extract Texts from Files

"repair" extracted texts:

- distorted diacritics in PDFs
- headers, footers, page numbers
- hyphenation

TextPerience I: Extract Texts from Files

selecting the right content from web pages:
JusText parameters vs. customization

Sekunda, která rozděluje svět.

Půlka lidí slyší něco jiného než vy

final class	good
cotext-free class	short
heading	True
length (in characters)	66
number of characters within links	0
link density	0.000
number of words	11
number of stopwords	5
stopword density	0.455
html.body.div.div.div.div.div.div.div.h1	

TextPerience II: Detect Language(s)

the `langid` module, Lui and Baldwin (2012)

- trained for $\approx$ 90 languages, excl. cestina
- Naive-Bayes classifier on letter n-grams
- good for longer texts
- noise reduction needed (HTML tags and entities, URLs, non-words etc.)
- how about multi-lingual documents?

TextPerience III: Keyword Search

"language independent" way: TF-IDF

- tokenize the text
- detect POS
- assume keywords to be only nouns
- lemmatize nouns
- compute TF-IDF on lemmata
- what about keyphrases?

TextPerience III: Keyword Search

Tagging and Lemmatization: a very simplistic approach

- find a word form in corpus (limit to 10 occurrences)
- save the most frequent lemma and tag
- works for all POS-annotated corpora with lemmatization
- fast and "independent" but not very precise

TextPerience III: Keyword Search

TF-IDF on noun lemmata

- corpora of different sizes for different languages
- precomputed d_fs in our corpora
- what about words not present in our corpora?

TextPerience IV: Named Entity Recognition

Stanford NER with different models, Finkel et al. (2005)

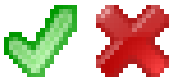
- ready to use models for es, en, de
- training data for cs and ru (see IA161) $\rightarrow$ new models
- very different model parameters for e.g. English and German
- apply models for a language on close languages (e.g. es model on fr)
- for training models, we need NE annotated corpus

TextPerience V: Wikipedia Search

use of API MediaWiki (2015)

- for each keyword/NE try to get the most relevant Wikipedia article
- no keyword scoring employed
- no graph-based scoring employed
- no generalization methods
- different sizes of Wikipedias in different languages

TextPerience VI: Evaluation

users can click on the evaluation icons 

- 443 files
- 1541 evaluations:
  - entities: 586 correct, 220 incorrect
  - keywords: 401 correct, 150 incorrect
- no learning from evaluations

Conclusion and Further Work

1. convert an uploaded file to text

   repair texts from PDFs

2. detect language

   detect multi-lingual documents, noise reduction

3. apply keyword search
   better ways to fast lemmatization
   keyword search in multi-lingual documents

   keyphrase search (what is a phrase?)

4. recognize named entities
   evaluation of existing models

   evaluation of using existing models on close languages

5. perform Wikipedia search of the keywords
   keyword weighting
   graph algorithms for disambiguation

   generalization methods (Wikipedia categories)

6. evaluate the results
   learning methods

Try It

Try TextPerience
`http://nlp.fi.muni.cz/projects/textperience`

References I

Finkel, J. R., Grenager, T., and Manning, C. (2005). 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, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lui, M. and Baldwin, T. (2012). Langid.py: An Off-the-shelf Language Identification Tool. In Proceedings of the ACL 2012 System Demonstrations, ACL '12, pages 25–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

MediaWiki (2015). API:Search — MediaWiki, The Free Wiki Engine.

Pomikálek, J. (2011). jusText. software.