Web Mining

Web Content Mining
- Part 2 -

Prof. Dr. Christian Bizer
Cäcilia Zirn
Robert Meusel

FFS 2014
Roadmap

- Part 2 -

- Information Extraction
  - KnowItAll

- Aspect-based opinion mining
  - Introduction
  - Frequent feature extraction
  - Opine
Information Extraction

定义

信息抽取（IE）的任务是自动从非结构化或半结构化的机器可读文档中抽取结构化信息。

文献

- 第30章：“信息抽取”

- 非常感谢Oren Etzioni和Zheng Shao提供的内容
Roadmap

- Part 2 -

- Information Extraction
  - KnowItAll

- Aspect-based opinion mining
  - Introduction
  - Frequent feature extraction
  - Opine
KnowItAll (Etzioni et al., 2004)

- A well-known information extraction system

- Goal: Extracting facts from the web
  - Populates ontology classes with instances
    - e.g. countries, singers, ...
    - IsA relations
  - Also binary relations like
    - PlaysFor(Athlete, SportsTeam)
    - CapitalOf(City, Country)
    - CeoOf(Person, Company)

- Input
  - set of classes
  - set of relations
  - generic rule templates
KnowItAll (KIA) (Etzioni et al., 2004)

Components:

- **Search Engine Interface**
  - Uses external search engines (Google, Altavista, ...)
  - KIA automatically creates query for extraction rules

- **Extractor**
  - Formulates extraction rules for each class and relation
  - Instantiated from set of generic templates

- **Assessor**
  - Assesses likelihood that extracted phrase is correct
  - by calculating PMI of extracted instance and associated phrases
  - statistics combined via Naive Bayes classifier

- **Database**
  - Stores information
Extraction Template Examples

**Instances of a class**
- NP₁ “such as” NPList₂
- NP₂ “and other” NP₂
- NP₁ “is a” NP₂

“Compact system cameras *such as* the Sony NEX-7 ...”
„The Panasonic Lumix and other compact system cameras...“

**Binary relations**
- NP₁ „plays for“ NP₂
- NP₁ „is located in“ NP₂

“Tim Oldenburg *plays for* Frankfurt Skyliners...”
„Paris *is located in* France...“
A noun phrase or nominal phrase (abbreviated NP) is a phrase which has a noun (or indefinite pronoun) as its head word, or which performs the same grammatical function as such a phrase.

„... the camera with the extraordinary resolution that I bought yesterday in the shop...“
Hearst Patterns (Hearst 1992)

Patterns

- such NP as \( \text{NP}_1, \text{NP}_2, \ldots \)
- NP such as \( \text{NP}_1, \text{NP}_2, \ldots \)
- \( \text{NP}_1, \text{NP}_2, \ldots \) and other NP
- NP, (especially | including) \( \text{NP}_1, \text{NP}_2 \) (or | and) \( \text{NP}_3 \ldots \)

“Compact system cameras including the Sony NEX-7 or the Olympus PEN...”

“such compact system cameras as the Sony NEX-7 ...”

Patterns were introduced to extract hyponymy relations
(= “is a” relations)
KnowItAll Extractor

- Applies patterns to search engine results
- Checks for heads of noun phrases

Noun phrase analysis
- A. “China is a country in Asia”
- B. “Garth Brooks is a country singer”

- In A, the word “country” is the head of a simple noun phrase.
- In B, the word “country” is not the head of a simple noun phrase.

- So, China is indeed a country while Garth Brooks is not a country.
KnowItAll Extractor

Rule Template:

- NP1 “such as” NPList2
  & head(NP1) = plural( name(Class1 ))
  & properNoun( head( each( NPList2 )))
  => instanceof( Class1, head( each( NPList2 )))

The Extractor generates a rule for “Country” from this template by substituting “Country” for “Class 1”.
Table 1: Three of the most productive rules for each class, along with the number of correct extractions produced by each rule, and the rule’s overall precision (before assessment).

<table>
<thead>
<tr>
<th>Rule</th>
<th>Correct Extractions</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>the cities of (&lt;city&gt;)</td>
<td>5215</td>
<td>0.80</td>
</tr>
<tr>
<td>headquartered in (&lt;city&gt;)</td>
<td>4837</td>
<td>0.79</td>
</tr>
<tr>
<td>for the city of (&lt;city&gt;)</td>
<td>3138</td>
<td>0.79</td>
</tr>
<tr>
<td>in the movie (&lt;film&gt;)</td>
<td>1841</td>
<td>0.61</td>
</tr>
<tr>
<td>(&lt;film&gt;) the movie starring</td>
<td>957</td>
<td>0.64</td>
</tr>
<tr>
<td>movie review of (&lt;film&gt;)</td>
<td>860</td>
<td>0.64</td>
</tr>
<tr>
<td>and physicist (&lt;scientist&gt;)</td>
<td>89</td>
<td>0.61</td>
</tr>
<tr>
<td>physicist (&lt;scientist&gt;)</td>
<td>87</td>
<td>0.59</td>
</tr>
<tr>
<td>(&lt;scientist&gt;), a British scientist</td>
<td>77</td>
<td>0.65</td>
</tr>
</tbody>
</table>
KnowItAll Assessor

- **Assessor filters result from extractor component**
  - Decides whether extracted information is plausible and dismisses it otherwise
  - Generates discriminator phrases for extracted entities
    - any phrase in which the extracted entities commonly appear together
    - phrases derived from the rules containing the entities
  - Uses the discriminator phrases as „features“ to classify the information as correct or false
Example

- Validate „France“ and „London“ as candidates for „country“
  
  - discriminator phrases:
    - „countries such as France“ / „countries such as London“
    - „the government of France“ / „the government of London“
    - „country ... France“ / „country ... London“

- Calculate Point-wise mutual information (PMI) of discriminator phrase and extracted information!
Know It all Assessor: Example of PMI

- **Discriminator**: “countries such as X”
- **Instance**: “France” vs. “London”
- PMI for France $\gg$ PMI for London (2 orders of magnitude)
- distinguish “high” PMI from “low” PMI for a discriminator

\[
PMI = \frac{27,800}{14,300,000} = 1.94E^{-3}
\]

\[
PMI = \frac{71}{12,600,000} = 5.6E^{-6}
\]
Assessor: Features from PMI

- Use thresholded PMI scores for classifier
- Learn conditional probabilities for PMI > threshold,
  given that the extracted entity is in the target class, or not

\[
P(\text{PMI} > \text{thresh} | \text{class})
\]

\[
P(\text{PMI} > \text{thresh} | \text{not class})
\]

- Train each discriminator separately
- Combine results of discriminator PMIs
Figure 11: Trained discriminators for the class city. Bootstrapping has learned a threshold on PMI scores that splits positive from negative training seeds, and has estimated conditional probabilities that the PMI score is above that threshold, given that the extraction is of the class or not of the class.
KnowItAll Assessor Bootstrapping

Bootstrap assessor

- Start with first extraction phase
- Use assessor, pick instances with highest PMI as seeds
- Train discriminators with seeds (positive and negative examples)
- select best discriminators
- use just those best discriminators to choose best seeds
The KnowItAll System

Domain-independent Rule Templates
<class> “such as” NP

Predicates
Country(X)

Bootstrapping

Extraction Rules
“countries such as” NP

Discriminators
“country X”

Extractor

Extractions
Country(“France”)

Assessor

Validated Extractions
Country(“France”), prob=0.999
Results for Unary Predicates

High precision and high recall for unary (instance of) extraction. More errors for Country (“Latin America”, “Iriquois nation”, etc).
High precision for both CapitalOf and CeoOf.
Able to find capitals of all cities.
Generic rules are too specific to find CEO’s for low-frequency companies.
Roadmap

- Part 2 -

- Information Extraction
  - KnowItAll

- Aspect-based opinion mining
  - Introduction
  - Frequent feature extraction
  - Opine
http://xkcd.com/937/
Objective: find what reviewers (opinion holders) liked and disliked

- Product aspects and opinions on the aspects

Since the number of reviews on an entity can be large, an opinion summary should be produced.

- Desirable to be a structured summary.
- Easy to visualize and to compare.
- Analogous to but different from multi-document summarization.
The tasks

- We have 5 tasks, but only focus on two.
  - Task 2 (aspect extraction and grouping): Extract all aspect expressions of the entities, and group synonymous aspect expressions into clusters. Each aspect expression cluster of entity $e_i$ indicates a unique aspect $a_{ij}$.
  - Task 4 (aspect sentiment classification): Determine whether each opinion on an aspect is positive, negative or neutral.
Roadmap

- Part 2 -

- Information Extraction
  - KnowItAll

- Aspect-based opinion mining
  - Introduction
  - Frequent feature extraction
  - Opine
Frequent feature extraction (Hu & Liu, 2004)

Tasks

1. Identifying product features
2. Extracting opinion words for features
3. [Producing a summary of the given information]

Not discussed here
1. Identifying product features

- **Intuition:** Important features are talked about by many customers

- **Approach**
  - reviews are POS-tagged
  - **nouns** and **noun phrases** are extracted as potential features
  - their **occurrence frequencies** are counted and only frequent ones are kept
  - threshold can be set experimentally
2a. Opinion words extraction

- POS-tags are used again
- For each frequent feature, the closest adjective is chosen as its opinion word

„The strap is horrible and gets in the way...“

„The horrible strap is attached to the camera...“

- In both examples, horrible will be the opinion word for strap.
Frequent feature extraction (Hu & Liu, 2004)

2.b Determine semantic orientation of opinion term

- Polarity of opinion term could be looked up in sentiment lexicon
- In the work of Hu & Liu, as a lexicon was not available, they looked for adjectives with known polarities in WordNet that are related to their extracted ones
Roadmap

- Part 2 -

- Information Extraction
  - KnowItAll

- Aspect-based opinion mining
  - Introduction
  - Frequent feature extraction
  - Opine
OPINE (Popescu & Etzioni, 2005)

Opine’s subtasks

- I. Identify product features.
- II. Identify opinions regarding product features.
- III. Determine the polarity of opinions.
- (IV. Rank opinions based on their strength.)

Based on KnowItAll!

Not discussed here
Opine

OPINE
Ana-Maria Popescu, Bao Nguyen, Oren Etzioni

Review Summary

Service quality: excellent (3), good (2), best, professional, better, view all (8)
Service attention: attentive (2)

Room beauty: absolutely beautiful, beautiful, view all (2)

User comments:
The service was excellent and our room was absolutely beautiful. Read more

When compared to Mandarin Oriental New York, Room beauty is
- worse at The Premier (33 others)

Quality: best, finest, love, better, view all (4)
Staff courtesy: extremely courteous, courteous, view all (2)
Beauty: beautiful
Room quality: gorgeous, complementary, view all (2)
Food quality: lovely, nice, view all (2)
Service discretion: discreet
Roadmap

- Part 2 -

■ Information Extraction
  ■ KnowItAll

■ Aspect-based opinion mining
  ■ Introduction
  ■ Frequent feature extraction
  ■ Opine
    - Feature extraction
    - Opinion phrase extraction
    - Sentiment orientation
Opine: Feature Extraction

- Product classes
  - Hotels

- Instances
  - Trump International

<table>
<thead>
<tr>
<th>Extracted Features</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Quality Size</td>
</tr>
<tr>
<td>Parts</td>
<td>Room</td>
</tr>
<tr>
<td>Features of parts</td>
<td>RoomSize</td>
</tr>
<tr>
<td>Related concepts</td>
<td>Neighborhood</td>
</tr>
<tr>
<td>Features of related concepts</td>
<td>NeighborhoodSafety</td>
</tr>
</tbody>
</table>

- OPINE also extracts opinion phrases
I loved the hot water and the clean bathroom.

The fan was broken and our room was hot the entire time.

I like a nice, hot room when the snow piles up outside.

Extract noun phrases np such that np contains only nouns and frequency(np)>1 as potential features.
I loved the hot water and the clean bathroom.

The fan was broken and our room was hot the entire time.

I like a nice, hot room when the snow piles up outside.

Assess potential features using bootstrapped lexical patterns (discriminators)

Examples
- X of Y
- Y has X
- Y’s X
- Y with X
- Y comes with X
- Y equipped with X
- Y contains X
- Y offers X
I loved the hot water and the clean bathroom.

The fan was broken and our room was hot the entire time.

I like a nice, hot room when the snow piles up outside.

Opine: Feature Extraction

Assess potential features using discriminators

\[
\text{PMI}(\text{hotel’s}[Y], \text{room}) = \frac{\text{hits(“hotel’s room”)}}{\text{hits(“hotel’s])*hits(“room”)}}
\]

\[
\text{PMI}(\text{hotel’s}[Y], \text{room}) = 0.54 \times 10^{-13}
\]

\[
\text{PMI}(\text{hotel’s}[Y], \text{snow}) = 0.64 \times 10^{-16}
\]

\[
\text{PMI}(\text{hotel’s}[Y], \text{room}) \gg \text{PMI}(\text{hotel’s}[Y], \text{snow})
\]
**Opine: Feature Extraction**

- I loved the hot **water** and the clean **bathroom**.

- The **fan** was broken and our **room** was hot the entire **time**.

- I like a nice, hot **room** when the snow piles up **outside**.

Assess potential features using discriminators

\[
\text{PMI}(\text{hotel’s}[Y], \text{room}) = \frac{\text{hits}(\text{“hotel’s room”})}{\text{hits}(\text{“hotel’s”}) \times \text{hits}(\text{“room”})}
\]

- \[\text{PMI}(\text{hotel’s}[Y], \text{room}) = 0.54 \times 10^{-13}\]

- \[\text{PMI}(\text{hotel’s}[Y], \text{snow}) = 0.64 \times 10^{-16}\]

- \[\text{PMI}(\text{hotel’s}[Y], \text{room}) \gg \text{PMI}(\text{hotel’s}[Y], \text{snow})\]
Roadmap

- Part 2 -

- Information Extraction
  - KnowItAll

- Aspect-based opinion mining
  - Introduction
  - Frequent feature extraction
  - Opine
    - Feature extraction
    - Opinion phrase extraction
    - Sentiment orientation
Opine: Opinion Phrases Extraction

- Opine uses extracted features to identify potential opinion phrases
- Analyses syntactic dependencies with MINIPAR
- Applies extraction rules to output of MINIPAR
- Opinion phrase consists of head word with its modifier
- Finally, semantic orientation of head is determined
Opine: Opinion Phrase Extraction

Given feature $f$, extract $po$ if:

$\exists po$ such that $pos(po) = adj|nn$, $mod(po,f)$

- $f = \text{feature}$
- $po = \text{potential opinion}$
- $pos = \text{part-of-speech tag}$
- $adj = \text{adjective}$
- $nn = \text{noun}$
- $mod = \text{modifies}$

„Given feature $f$, extract $po$ if

- there is a $po$ with its pos-tag being $adj$ or $nn$
- and $po$ modifies $f$

„I love its $great$ resolution“, „and need fast $autofocus$“
Opine: Opinion Phrase Extraction

Given feature $f$, extract $po$ if:
$\exists po$ such that $pos(po) = adv$, $\exists (S, O)$ such that $S = f$, $O = po$

- $f = \text{feature}$
- $po = \text{potential opinion}$
- $pos = \text{part-of-speech tag}$
- $adv = \text{adverb}$
- $S = \text{subject}$
- $O = \text{object}$

„Given feature $f$, extract $po$ if
- there is a $po$ with its pos-tag being $adv$
- $f$ is the $subject$ and $po$ is the $object$

„The scanner works well“ „the camera shoots fast“
Why Parsing?

- For those rules, POS-tagging is not sufficient
- Deeper syntactic analysis of sentence required
- Analysis of
  - subject, predicate, object
  - modifying relations
    („great resolutions“ -> resolutions is modified by great)
  - extract heads of noun phrases
  - ....

Just a side note: Stanford CoreNLP includes a parser!
Natural language parsing is the task of analyzing the syntactic structure of a sentence.

Syntactic structure of language is defined by the grammar.
Parsing Techniques

- Parsing techniques known from compiler construction (PI2)
- Parsers take grammar rules as input
  - hand-written or learned from manually labelled data
- Basic strategies
  - Top-down parsing: Start with start symbol, apply rules until you receive input sentence
  - Bottom-up parsing: Starts with the input sentence (leaves of the tree), find rules to reduce tree to the start symbol.

<table>
<thead>
<tr>
<th>Easy Grammar</th>
</tr>
</thead>
<tbody>
<tr>
<td>S -&gt; NP VP</td>
</tr>
</tbody>
</table>
| NP -> NP CC NP | Det Adj N | /
| VP -> V NP |
| Det -> the |
| N -> room | hotel |
| V -> loved |
| CC -> and |
MINIPAR (used by Opine)

- [http://gate.ac.uk/sale/tao/splitch17.html#sec:parsers:minipar](http://gate.ac.uk/sale/tao/splitch17.html#sec:parsers:minipar)

- **Shallow parser**
  - does not build complete tree
  - analyses dependency relations between words

- For each word, the output is
  - the **head** modified by this word
  - name of **the dependency relationship** between this word and the head

- By using MINIPAR, Opine receives the necessary information to apply the opinion phrase extraction rules.
Roadmap

- Part 2 -

- Information Extraction
  - KnowItAll

- Aspect-based opinion mining
  - Introduction
  - Frequent feature extraction
  - Opine
    - Feature extraction
    - Opinion phrase extraction
    - Sentiment orientation
Opine: Semantic Orientation

The room was hot(-) and stuffy(-).

After freezing for hours, the room was nice(+) and hot(+).

cold basic loud visible casual modern central quiet

- After the potential opinion phrases are extracted, OPINE assigns their heads one of 3 semantic orientation labels (positive, negative, neutral)

- More formally, OPINE computes a SO label for a word in the context of a product feature and a sentence. For example, hot is negative...

- Initial scores of phrases can be derived from a sentiment lexicon
Task: Compute the SO label for a (word, feature, sentence) tuple

OPINE solves the task in 3 steps.

- An overall SO label for a word is computed.
  - SO(word)
- A SO label for a word in the context of a given feature is computed
  - SO(word, feature)
- A SO label for a word in the context of a given feature and a given sentence is computed
  - SO(word, feature, sentence)

Each solution step = labeling problem \(\rightarrow\) relaxation labeling
Opine: Relaxation Labeling

Unsupervised classification technique

Input

- Set of objects (e.g. words)
- Set of labels (e.g. SO labels)
- Initial probabilities for each object’s possible labels
- Definition of an object’s neighborhood (other objects)
- Definition of neighborhood features
- Definition of support function for object label

Positive, negative, neutral

E.g. the word “nice“ participates in conjunction “and“ together with another word whose label is estimated positive
Opine: Relaxation Labeling

- Used when the label of a given object is constrained by the labels of other objects (its neighborhood)
  - Here: label of a word is influenced by
    - other words attached to it in the sentence
    - by the known labels of synonymous words
    - ...

- The influence of an object’s neighborhood on the object’s label is quantified by a support function.

- Starts with an initial assignment of labels to objects and iteratively modifies this assignment.
  - At each iteration, it updates the probability of each label of each object based on current probability and on the current labels of the object’s neighbors.

- RL stops when some termination criterion is met (e.g. when global label assignment stays constant)
Building word neighborhoods:

- conjunctions, disjunctions
- syntactic attachment rules
- WordNet synonymy/antonymy
- morphology information
Opine: Results

Dataset (Hu&Liu’04)
- 5 consumer electronics product classes
- 314 reviews

Results Extraction
- $P = 94\%$  $R = 77\%$

Results Polarity
- $P = 78\%$  $R = 88\%$
Next (and last) exercise:
Tuesday, 09.04.2013 (after easter holidays)

You will try some simple feature extraction and look up their polarities in a sentiment lexicon
Organizational issues: Project work

- During the easter holidays, please think of a possible project
- Project should be related to one of the three topics of the course
  - Web usage mining
  - Web structure mining
  - Web content mining
  - Exceptions are possible, but the topic should then be at least related to data mining!
- Data sets for the project should be extracted from the web, and if possible generated by the community