Text Mining

2004-2005
Master TKI

Antal van den Bosch en
Walter Daelemans
http://ilk.uvt.nl/~antalb/textmining/

Dinsdag, 10.45 - 12.30, SZ33

Timeline (1)

- [1 februari 2005]
  - Introductie (WD)
- [15 februari 2005]
  - Syntactic pipeline 1: Tokenization, POS tagging (AB)
- [22 februari 2005]
  - Concept chunking (Sander Canisius)
- [1 maart 2005]
  - Syntactic pipeline 2: chunking, relation finding (WD)

Timeline (2)

- [8 maart 2005]
  - Named-entity recognition (Toine Bogers)
- [15 maart 2005]
  - Information extraction (WD)
- [5 april 2005]
  - Tools (AB)
- [12 april 2005]
  - Industrial information extraction (Martijn Spitters, Textkernel B.V.)

Timeline (3)

- [19 april 2005]
  - Information extraction from spoken user input (Piroska Lendvai)
- [26 april 2005]
  - Ontology learning (Marie-Laure Reinberger)
- [3 mei 2005]
  - Factoids (AB)
- [10 mei 2005]
  - Presentaties

Overview

- The syntactic pipeline (1)
  - Tokenization
    - What is a token?
    - General and special tokenizers
  - PoS tagging
    - (work of Jakub Zavrel, Walter Daelemans, Hans van Halteren, 1996-1999)
    - What is PoS tagging?
    - The CGN case
    - Lemmatization

Tokenization

- What is a token?
  - A delimited string of characters
  - Delimiters separate tokens
  - Delimiters:
    - “white space” (spaces, tabs, newlines)
    - punctuation
    - markup (SGML, HTML, XML, …)

Tokenization

What is a token?

- <sentence>
- What
- is
- a
- token
- ?
- </sentence>

Tokenization: main problem

- Punctuation sometimes belongs to the word
  - nitty-gritty
  - abbr.
  - President J.F. Kennedy
  - (semi-)ironic
  - the "F*"-word

(Incomplete) solutions

- Abbreviation lists
  - Language specific
  - Domain specific
- Word grammars
  - Regular expressions
  - Language specific
- Punctuation conventions/habits
  - Language specific

More tokenization issues

- Contracted forms:
  - don't = do not?
  - I'll = I will?
- White space also meaningful?
  - Double newline
- Typesetting features (bold, italics, font size) also meaningful?

Sentence splitting

- "sentence tokenization"
- Sentence ≈ syntactic domain
- Most European languages:
  - period, !, ?, end sentence (w/ rules for quotes)
  - first word is capitalized
- But:
  - Sentence may not end nicely
  - Other words are capitalized as well (names, nouns in German)
  - ¿Spanish?

Existing tokenizers

- Regexp-based
  - Commercially available
  - Many more custom tokenizers
- Learning tokenizers
  - Combined tokenizing/sentence splitting, stochastic, using PoS (Mikheev, 2000)
  - Sentence splitting (memory-based: Stevenson & Gaizauskas, 2000; maxent: Reynar & Ratnaparkhi, 1997)
  - Learning punctuation on transcribed speech via prosody (Christensen, Gotoh, Renals)

Overview

- The syntactic pipeline (1)
  - Tokenization
    - What is a token?
    - General and special tokenizers
  - **PoS tagging**
    - What is PoS tagging?
    - The CGN case

Part-of-speech tagging

- What is PoS tagging?
- Historical overview
- The CGN Case
  - Ensembles of classifiers
  - Bootstrapping a tagger for a new corpus
- PoS and lemmatization

POS tagging

- Assigning morpho-syntactic categories (Parts-of-speech) to words in context:

The	green	train	runs	down	that	track	.
Det	Adj/NN	NNS/VBZ	NN/VB	Prep/Adv/Adj	SC/Pron	NN/VB	.
Det	Adj	NNS	VB	Prep	Pron	NN	.

- Disambiguation: a combination of lexical and "local" contextual constraints.

POS tagging: what for?

- shallow processing (abstraction from words: >recall)
- basic disambiguation (choose form: >precision)
- robustness, coverage, speed.
- good enough for many applications:
  - text mining: information retrieval/extraction
  - corpus queries (linguistic annotation)
  - terminology acquisition
  - text-to-speech
  - spelling correction

POS: remaining errors

- last 10-3% is hard:
  - long distance dependencies
  - genuine ambiguities
  - annotation errors
  - unknown words
  - not enough information in the features
  - more features are needed, but this has an exponential effect on data sparseness.
  - generalization to general text is poor: 97% → 75%.
  - some languages: large tag sets & small corpora.

POS tagging in CGN

- Hand-annotate all 10M words
- ML-assisted
- Four taggers:
  - Hidden Markov modelling
  - Transformation-based learning
  - Maximum entropy modelling
  - Memory-based learning
- Bootstrapping on non-CGN data

Hidden Markov Modelling

- “tag sequence emits word sequence”
- Given a sequence of words, what is the most probable tag sequence?
- States are tags; $P_{transition} = P(t_i|t_{i-1})$
- $P_{emission} = P(w_i|t_i)$
- highest probability state sequence: Viterbi search.

Hidden Markov Modelling(2)

- Advantages:
  - Fast tagging and training
  - Easy to implement
  - Global optimization
- But:
  - sparse data: zero probabilities → smoothing (add-one, Good-Turing, interpolation, back-off).
  - more features: trigrams, context words?
  - unknown words: equiprobable or external guesser?

(Error-driven) Tranformation-based Learner

- General idea: (Brill, 1994)
  start with base annotation, and perform error-reducing greedy search for transformation rules (exhaustive, but data driven).
- Separate learner for unknown words and contextual rules.
- Base annotation:
  - known words are assigned their most likely part of speech,
  - unknown words are tagged NP if capitalized, NN otherwise.

Tranformation-based Learner

- Advantages:
  - more complicated features than HMM
  - Produces concise and intelligible rule-set
  - fast tagging
- But:
  - no probabilities
  - slow training

Maximum Entropy Modelling

General idea: (Ratnaparkhi, 1996)

- Tagging, as a classification task, can be solved by combining diverse forms of contextual information in a probabilistic model.
- Maximum Entropy: “model all that is known and assume nothing that is unknown”.

Maximum Entropy Modelling

- Probability model assumes anything as a binary feature with its own weight
- Generalized Iterative Scaling algorithm searches for a model that:
- observes the constraints expressed by the features and the data.
- has the maximum entropy. This model is unique and GIS will converge to it.

Maximum Entropy Modelling

- Advantages:
  - Easy integration of different features
  - Model gives accurate probabilities
  - MaxEnt weights take feature correlation into account
  - Each value of a feature has its own weight.
- But:
  - Low-frequency data must be discarded to avoid overfitting.
  - Training is quite slow.

Memory-Based Learning

- (Daelemans et al., 1996)
  Similar situations have similar outcomes.
- Tagging = a classification task solved by similarity-based reasoning from labeled examples stored in memory
- Straight analogical reasoning

=	=	John	will	join	np
=	John	will	join	the	md
John	will	join	the	board	vb
will	join	the	board	=	dt
join	the	board	=	=	nn

Memory-Based Tagger construction

- Initial lexical representations: Construct frequency-sensitive ambiguous category lexicon. Percentual threshold (e.g. 10).
- A case base for known words is constructed:
- A case base for unknown words is constructed:
- MBTs are constructed for the two case-bases.

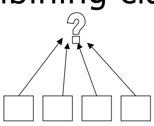
Memory-Based Learning

Advantages:
- Easy combination of different features
- Robustness against overfitting.
- Fast training and tagging with igtree

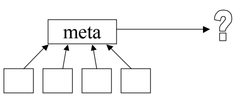
But:
- Weighting does not look at feature correlations, and averages over all feature values
- No global optimization (yet)
- Trade-off between speed and accuracy

Combining classifiers

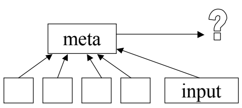
- Voting



- Stacking



- Arbiter



Combining classifiers

- Voting: democratic. Ties?
- Stacking: assign weights to votes on basis of reliability/error
- Arbiter:
  - Stacking advantage
  - Recognize reoccurring errors,
  - Correct voters

Bootstrapping from existing resources

- Problem setting: POS tagging with a new tagset (CGN) with very small training corpus.
- Stacking/arbiter allows including components that use other tagsets, e.g. existing taggers and lexicons.
- Can a meta-learner use a very small training set to learn the mapping?

Bootstrapping experiments

- We trained 4 taggers on small samples of DutchCGN corpus (TnT, MBT, Brill, MXPOST)

Basic results	5000			10000			20000		
(a la van Halteren)	u	k	t	u	k	t	u	k	t
MBT	39.4	90.8	82.0	46.3	91.6	85.4	45.9	93.0	88.3
TNT	49.0	91.8	84.5	50.0	92.2	86.4	57.4	94.5	90.8
MAX	50.0	79.5	74.4	58.1	86.2	82.4	57.4	90.4	87.0
RUL	29.8	87.7	77.7	37.5	87.5	80.7	40.2	89.7	84.7
CGN ensemble			84.3			87.2			90.5
% unknown			17.2			13.7			10.1

Bootstrapping experiments

Available resources: CELEX, Word, 9 WOTAN taggers (Wall):

	5000	10000	20000
CGN Ensemble	84.3	87.2	90.5
CGN + Word	83.7	87.6	90.5
CGN + CEL	85.6	88.2	91.2
CGN + Wall	91.3	91.4	93.4
Word	73.1	75.6	80.1
CEL	25.7	27.4	29.5
Wall	90.1	91.0	91.5
CGN + Wall + CEL + Word	91.4	91.7	93.5
error reduction	-44.7	-39.0	-29.6

Algorithm combination

- Produce output for each classifier for each data item by 10-fold cross validation.
- (=use 90% for training and 10% for testing)
- Combination methods: majority voting, stacking, arbitering (meta-learner: IB1, IB1-IG, MVDM).
- Compare with best single algorithm

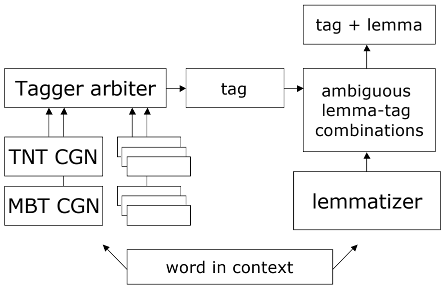
Algorithm combination

	unknown	known
best (maccent)	83.2	98.1
majority	84.7	98.3
combi	85.1	98.4
arbiter	86.4	98.6

CGN continues

datum	nov99	feb00	mar00	jul00	jan01	feb02	may02	feb03
TnT	**89.1**	**91.6**	**92.7**	**93.9**	**95.3**	**96.2**	**96.4**	**96.8**
MBT	**86.5**	**89.4**	**91.2**	**92.0**	**94.3**	**95.6**	**95.9**	**96.3**
Maxent	**83.6**	**89.4**	**90.1**	**92.6**	**95.2**			
Brill	**83.3**	**86.3**	**87.9**	**89.9**				
Arbiter	**94.2**	**94.3**	**94.3**	**95.6**	**96.2**	**96.6**	**96.8**	**97.1**
# words	10802	21475	39304	95246	553226	2762712	3612845	6049752

CGN Lemmatizer



Memory-based lemmatizer

- Input: word (*boek*)
- Output: for all possible lemmatizations,
  - POS tag (*N* or *V*)
  - Spelling change *(no* or +*en*)
- Train on CGN lexicon
- Exact lookup of known words
- (Van den Bosch & Daelemans, 1999)

Memory-based lemmatizer

- Examples:

boek	N(12)|WW(16)+Ien
bestal	WW(19)+Dal+Ielen
genen	N(16)+Den|VNW(12)+Dn
amnesie	N(13)
databases	N(16)+Ds

- 93% precision, 91% recall of POS+lemma for *unknown* words