IT MAKES SENSE

A Wide-Coverage
Word Sense Disambiguation System
for Free Text

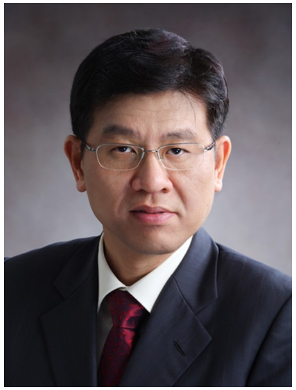
Zhi Zhong & Hwee Tou Ng

Zhong Zhi

- Mysterious researcher
- Email: hongzhi@comp.nus.edu.sg
- Should I drop her (him?) an email and ask “who are you?”

Hwee Tou NG

- Provost's Chair Professor
- Department of Computer Science,
  School of Computing,
  National University of Singapore
- Email: nght@comp.nus.edu.sg
- Home Page:
  http://www.comp.nus.edu.sg/~nght



Word Sense Disambiguation?

- There is a cup on the table.

Word Sense Disambiguation

- There is a **<u>cup</u>** on the **<u>table</u>**.

?

?

- There is a **cup** on the **table**.

1) (N) a small open container usually used for drinking; usually has a handle
2) (N) the hole (or metal container in the hole) on a golf green
3)....

1) (N) a set of data arranged in rows and columns
2) (N) a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs
3)....

Word Sense Disambiguation

- There is a **cup** on the **table**.

cup:

1) (N) a small open container usually used for drinking; usually has a handle
2) (N) the hole (or metal container in the hole) on a golf green
3)....

table:

1) (N) a set of data arranged in rows and columns
2) (N) a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs
3)....

How does Hwee Tou's group solve this problem?

Intuition

- Collect sense-annotated text
- Extract features
- Use machine learning algorithms to learn the relation between a word features and its tagged sense
- With a given word and its features, one can use the learnt function to predict the relevant sense

- Three steps
  - Pre-processing
    - To get auto-annotated text
  - Feature & Instance Extraction
    - Extract word features (POS, surrounding words, local collocations)
  - Classification
    - SVM to learn the mapping between words' features and senses

Preprocessing

- Detect sentence boundaries in a raw input text with a sentence splitter.
- Tokenize the split sentences with a tokenizer
- POS tagging
- Lemmatize each words

- I don't think a computer can understand human language. Does it make any sense?

- I don't think a computer can understand human language. / Does it make any sense?
  =>
  S1: I don't think a computer can understand human language.
  S2: Does it make any sense?

- S1: I | do | n't | think | a | computer | can | understand | human language | .
- S2: Does | it | make | any | sense |?

- S1: I/PRP | do/VBP | n't/RB | think/VB | a/DT | computer/NN | can/MD | understand/VB | human/JJ language/NN | ./.
- S2: Does/NNP | it/PRP | make/VB | any/DT | sense/NN |?/.

- S1: I/PRP | do/VBP | n't/RB | think/VB | a/DT | computer/NN | can/MD | understand/VB | human/JJ language/NN | ./.
- S2: **Do/NNP** | it/PRP | make/VB | any/DT | sense/NN |?/.

Feature & Instance Extraction

- Feature 1: POS tags of surrounding words
  - Three words to the left
  - Three words to the right
  - The target word itself

Feature & Instance Extraction

- Feature 2: Surrounding words
  - Can be in the current sentence or immediately adjacent sentences
  - Stop words, words without alphabetic characters (punctuation, symbols, numbers, etc.) are removed
  - E.g:
    - All possible neighbours of the word “NLP” => [human, computer, algorithm, machine_translation]
    - Context: [computer, algorithm, computer]
    - Feature vector: [0,1,1,0]

- Feature 3: Local Collocations
  - Use 11 local collocations:
  - $C_{-2,-2}, C_{-1,-1}, C_{1,1}, C_{2,2}, C_{-2,-1}, C_{-1,1},$
  - $C_{1,2}, C_{-3,-1}, C_{-2,1}, C_{-1,2}, C_{1,3}$
  - $C_{i,j}$ = ordered sequence of words in the same sentence of word *w* (at 0)
  - i/j = starting/ending position
  - Negative/positive offset = Left/right

Feature & Instance Extraction

There	is	a	**cup**	on	the	table	.
-3	-2	-1	**0**	1	2	3	4

- $C_{-2,-2}$=”is”
- $C_{-2,-1}$=”is a”
- $C_{3,4}$=”table .”

Classification

- Each word has a classifier (sense predictor)
- The models are trained using supervised learning methods (SVM).
- Given a word, if its classifier exists, the results as a set of ordered pairs <$sense_i$, $prob_i$> will be returned
- If the word classifier doesn't exist, return predefined default sense.
- Else return “U”

- The performance of WSD system greatly depends on the size of training data used.
- Q: Where do they find big sense-annotated data?
- A: SEMCOR + DSO corpus + auto-generated from parallel texts

Auto-generated Training Data

- Used six English-Chinese parallel corpora (from Linguistic Data Consortium - LDC)
  - Hong Kong Hansards
  - Hong Kong News
  - Hong Kong Laws
  - Sinorama
  - Xinhua News
  - English translation of Chinese Treebank

Auto-generated Training Data

- Perform tokenization on the English text with Penn TreeBank tokenizer
- Perform Chinese word segmentation on the Chinese text (Low et al. 2005)
- Perform word alignment using GIZA++
- Assign Chinese translations to each sense of an English word w.

Auto-generated Training Data

- Pick the occurrences of w which are aligned to its chosen Chinese translations in the word alignment output of GIZA++
- Identify the senses of the selected occurrences of w by referring to their aligned Chinese translations.

Auto-generated Training Data

- Only extract top 60% most frequently occurring polysemous content words in Brown Corpus
  - 730 nouns
  - 190 verbs
  - 326 adjectives
  - 28 adverbs

Auto-generated Training Data

- For each of the top 60% nouns & adjectives, maximum of 1,000 training examples are gathered from parallel texts.
- For each of the top 60% verbs, not more than 500 examples from parallel text and not more than 500 examples from DSO corpus are collected.
- All data from SEMCOR

Auto-generated Training Data

- More than 21,000 classification models was generated.
- On average, each word has 38 training instances
- Total size of the models is ~200 MB

POS	NOUN	VERB	ADJ	ADV
# of types	11,445	4,705	5,129	28

Evaluation

	SensEval-2	SensEval-3
IMS	**65.3%**	72.6%
Rank 1 System	64.2%	**72.9%**
Rank 2 System	63.8%	72.6%
Most frequent sense	47.6%	55.2%

Conclusion

- Give much better compare to selecting most frequent sense.
- State-of-the-art WSD system
- Licensed in GPL?
  - Available for research
  - **NOT** for commercial use

Used Tools

- OpenNLP toolkit
  - Sentence splitter & POS tagger
  - http://opennlp.sourceforge.net
- Penn TreeBank tokenizer
  - http://www.cis.upenn.edu/~treebank/tokenizer.sed
- jWordnet (Lemmatization)
  - http://jwordnet.sourceforge.net
- Machine learning
  - LIBLINEAR is used by default
  - WEKA, LIBSVM & MaxEnt are also supported.

Want more?

- Yes
- No

Thank you

Q & A

Yes? Machine Learning

- How do machines learn?
  - Are fed with **data**
  - Detect **patterns** from data
- Type of learning
  - Supervised learning
    - Input & output are provided
  - Unsupervised learning
    - Variations of clustering (grouping)

- We collect a lot of data
- The machine will compare the data instances and group them into groups or organise them on a space.

- We all know this

  $y=f(x)$

  - Given a x and a function f, we can find y
- What if we have x, but function f is too complicated to define or we don't have it?
- Give up!

Nah, just kidding

- We may collect data instead
- <x1,y1>, <x2, y2>, <x3, y3> ...
- Try to fit a known function into this dataset
- We have a g(x) ~ f(x)
- Use g(x) instead of f(x)

- So in order to use most of the state-of-the-art supervised learning methods, you need to:
  - Collect data
  - know the form of the inputs and outputs
  - Convert the collected data into that form
  - Find a machine learning tool and use it (or make one by yourself)

Some suggestions

- Words as input
  - Lexicon: [dog, cat, fish, rabbit]
  - Dog = [1,0,0,0] or 1
  - Fish = [0,0,1,0] or 3
- Categorical output
  - LUK = [yes, no, unknown]
  - Yes = [1,0,0] or 1
  - No = [0,1,0] or 2
  - Unknown = [0,0,1] or 3
- Confident as output
  - Real number value between 0 and 1

Thank you

Q & A

- When was the paper published?
  - What did they react to?
- Why did they use SVM?
- Why did they use WordNet 1.7.1?
- How can we make this better?

Thank you