EVALITA 2011

Evaluation of NLP and Speech Tools for Italian

EVALITA 2011

The Tanl tagger for Named Entity Recognition on Transcribed Broadcast News

G. Attardi, G. Berardi, S. Dei Rossi, M. Simi
Dipartimento di Informatica, Università di Pisa

Introduction

- Annotating NEs
  - Special case of chunking
- Tanl chunker
  - Flexible and customizable tagging tool
    - PoS tagging, SuperSense tagging, …
  - Maximum Entropy classifier for learning how to chunk texts
  - Dynamic programming in order to select sequences of tags with the highest probability

Tanl Chunker

- Features Types
  - Attributes Features
    - Attributes like PoS, Lemma of surrounding tokens
    - relative positions w.r.t. to the current token
    - E.g: POSTAG -1 0
  - Local Features
    - Binary morphological features extracted from the analysis of the current word and the context in which it appears
    - E.g.: “previous word is capitalized”
  - Global Features
    - Properties holding at the document level
    - E.g.: if a word in a document was previously annotated with a certain tag
- Textual configuration file to specify features

Tanl Chunker

- Dictionaries
  - Used to group tokens with specific properties
  - Created automatically by pre-processing the training data
    - **Dictionary:** all annotated NEs that appear more than 5 times
    - **Prefix:** 3 letters prefixes of NEs
    - **Suffix:** 3 letters suffixes of NEs
    - **LastWords:** Words occurring as last in a multi-token entity
    - **FirstWords:** Words appearing as first in a multi-token entity
    - **LowerIn:** Lowercase words occurring inside an entity
    - **Bigrams:** All bigrams that precede an entity
    - **FrequentWords**
    - **Designators:** Words that precede an entity
- The tagger extracts use the dictionaries to extract binary some features
  - E.g.: suffix is present in Suffix dictionary, token is present in LastWords, …

Dataset

- Composed of 3 different corpora
  1. Broadcasts news manually transcribed and annotated with NEs
  2. The automatic transcription of the same news (without NEs)
  3. I-CAB: a corpus of (written) news stories annotated with NEs
- Only corpora 1 and 3 contain NEs and could be used for training purposes
- Problems
  - Different origins
  - Representative of quite different genres
- Corpus 1: No punctuation and sentence boundaries

Baseline

- Training corpus 1 divided into 2 sub-sets
  - 90% for training and 10% for development
- Basic configuration
  - No attributes features
  - Standard set of local features
    - Features of Current Word: first word of sentence and capitalized; first word of sentence and not capitalized; two parts joined by a hyphen
    - Features from Surrounding Words: both previous, current and following words are capitalized; both current and following words are capitalized; both current and previous words are capitalized; word is in a sequence within quotes
- 100 iterations of the Maximum Entropy algorithm
- F-score of 60.48

Tuning

- Added PoS column to Corpus 1
  - Hunpos Tagger trained on the corpus “La Repubblica”
- Creation of many configuration files with different combination of features
  - Different permutations of the attributes features involving POSTAG, CPOSTAG (first letter of the POSTAG) and NETAG
  - Variation of other parameters
    - Number of iterations
    - Cutoff feature
    - Refine feature
  - Evaluation based on a k-fold cross validation (k = 10)
  - Best run on the development set
    - F-score: 68.50
    - Configuration used for Run Closed 2

Run Closed 2

- Tanl Chunker
- Standard set of local features
- Attributes Features

POSTAG	-1 0 1
CPOSTAG	0
NETAG	-1

- Other parameters
  - Iteration: 100
  - Cutoff = 0
  - Refine enabled
    - IOB2 annotation split into a more refined set of tags
    - Helps the classifier to better separate the data

Run Closed 1

- Different approach: Stanford CRF-Classifier
  - Based on the Conditional Random Fields (CRF)
  - Gibbs sampling instead of other dynamic programming techniques for inference on sequence models
  - Works quite well using only the FORM column
    - Useful since the system output of the PoS tagger can contain errors
  - Two different models were created
    1. Using the full-set of tags in the IOB2 notation (a total of 8 classes)
    2. Using only the four semantic classes (not considering the prefixes 'B-' and 'I-')
  - Results analysis: on the development set the first model worked better on GPE and LOC, while the second one on ORG and PER
    - Outputs were combined to improve the overall performances

Open subtask - Run 1

- Added SuperSenses to the broadcast news corpus (Corpus 1)
  - model trained on the ISST-SST corpus (~300.000 tokens)
  - Three of the SuperSenses describe semantic classes similar to the NEs of this task
    - noun.location (LOC|GPE)
    - noun.person (PER)
    - noun.group (ORG)
- SuperSenses used as attributes feature to help the NE tagger to isolate and identify the entities
  - After some tuning, the best results were obtained with the same settings of Run Closed 2 and with the following attributes features

```
FORM      0
POSTAG    -2 -1 0 1 2
CPOSTAG   -2 -1
SST       0
NETAG     -2 -1
```

Open subtask - Run 2

- Created from the output of Run Closed 1
- Some post-processing heuristics were applied
  - NEs tag dictionary extracted from the corpus itself
  - SuperSenses from ItalWordNet (IWN)
  - Algorithm:
    - For each capitalized token, returns the most common NE tag associated to the token from the self extracted dictionary if available, otherwise returns the most common SuperSense from the IWN dictionary, converted to the corresponding NE tag

...and I-CAB?

- Many experiments using as training set the I-CAB 2009 corpus (~220.000 tokens) in addition to the broadcast news corpus (~40.000 tokens)
  - GOAL: give more training examples to the tagger
- Basic idea
  - Remove all punctuation and sentence boundaries from I-CAB to make it more similar the other corpus
- The results obtained using both corpora were worst with respect to the ones obtained with only the broadcast news corpus despite its small size
  - All the final runs produced with models trained only on the broadcast news corpus

Final results

	Accuracy	Precision	Recall	FB1
UniPI - run closed 1	95.59%	61.61%	47.23%	53.47
UniPI - run closed 2	95.64%	64.48%	50.45%	56.61
Best closed (A_1)	n.a.	61.70%	60.20%	60.94
UniPI - run open 1	95.85%	65.90%	52.09%	58.19
UniPI - run open 2	85.45%	54.83%	49.72%	52.15
Best open (A)	n.a.	65.25%	61.45%	63.29

…and on the gold test set:

	Accuracy	Precision	Recall	FB1	FB1 Impr.
UniPI - run closed 1	97.64%	78.17%	71.29%	74.57	+21.10
UniPI - run closed 2	97.14%	74.14%	69.88%	71.95	+15.34
UniPI - run open 1	97.45%	76.34%	72.75%	74.50	+16.31
UniPI - run open 2	97.04%	64.90%	70.46%	67.57	+15.42

Discussion

- Main difficulty: test set automatically extracted by the ASR system
  - Many transcription errors
  - Lacks of punctuation and sentence boundaries
  - Capitalization of words is not complete
  - Different from the training set, which was manually revised
- The results obtained on the runs are quite low considering the F-score, but the accuracy values are good
  - Hard to identify entities within the text stream without any marker, like capital letters, to indicate their presence
    - On the development set (a portion of the training corpus) and on the manually corrected test set the results were much higher
      - F-score about 15-20 points higher
      - All the relevant capital letters were manually added in the corpus
    - Heuristic used in Run Closed 2 failed for the same reason
- Our system is weak in dealing with the inaccuracies introduced by the ASR system