Named Entity Recognition through Corpus Transformation and System Combination

José A. Troyano

Dep. Computing Languages and Systems

University of Seville (Spain)

Index

- Problem, tools and resources
- Corpus transformation
- System combination
- Conclusions and future work

Named Entity Extraction and Recognition (I)

Named Entity Extraction (NEE): Subtask of Information Extraction (IE) that implies the extraction of proper names and their classification according to a taxonomy.

NEE = NER + NEC

Named Entity Recognition (NER): it is the identification of the word sequence that conforms the name of an entity.

Named Entity Classification (NEC): it is the subtask in charge of deciding which category is assigned to a previously recognized entity.

Named Entity Extraction and Recognition (II)

El presidente del COI, Juan Antonio Samaranch, ...

NEE

El	O
presidente	O
del	O
COI	B-ORG
,	O
Juan	B-PER
Antonio	I-PER
Samaranch	I-PER
,	O
...	...

NER

El	O
presidente	O
del	O
COI	B
,	O
Juan	B
Antonio	I
Samaranch	I
,	O
...	...

IOB notation

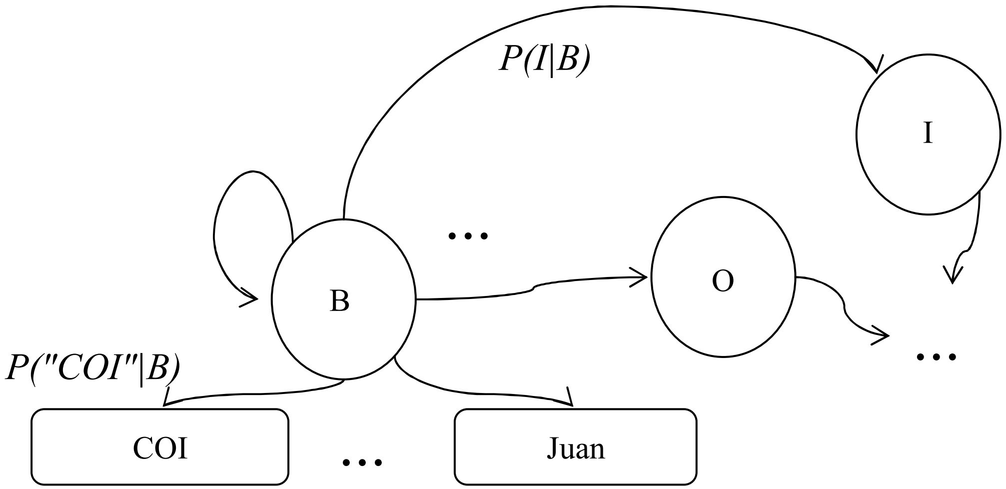
Taxonomy: PER, LOC, ORG, MISC

The corpus

We have used the Spanish corpus distributed for the Named Entity Recognition shared task of CoNLL-02:

- Training corpus: about 250000 tokens and 20000 entities.
- Test-A corpus: about 50000 tokens and 4000 entities.
- Test-B corpus: about 50000 tokens and 3500 entities.

TnT. Markov models

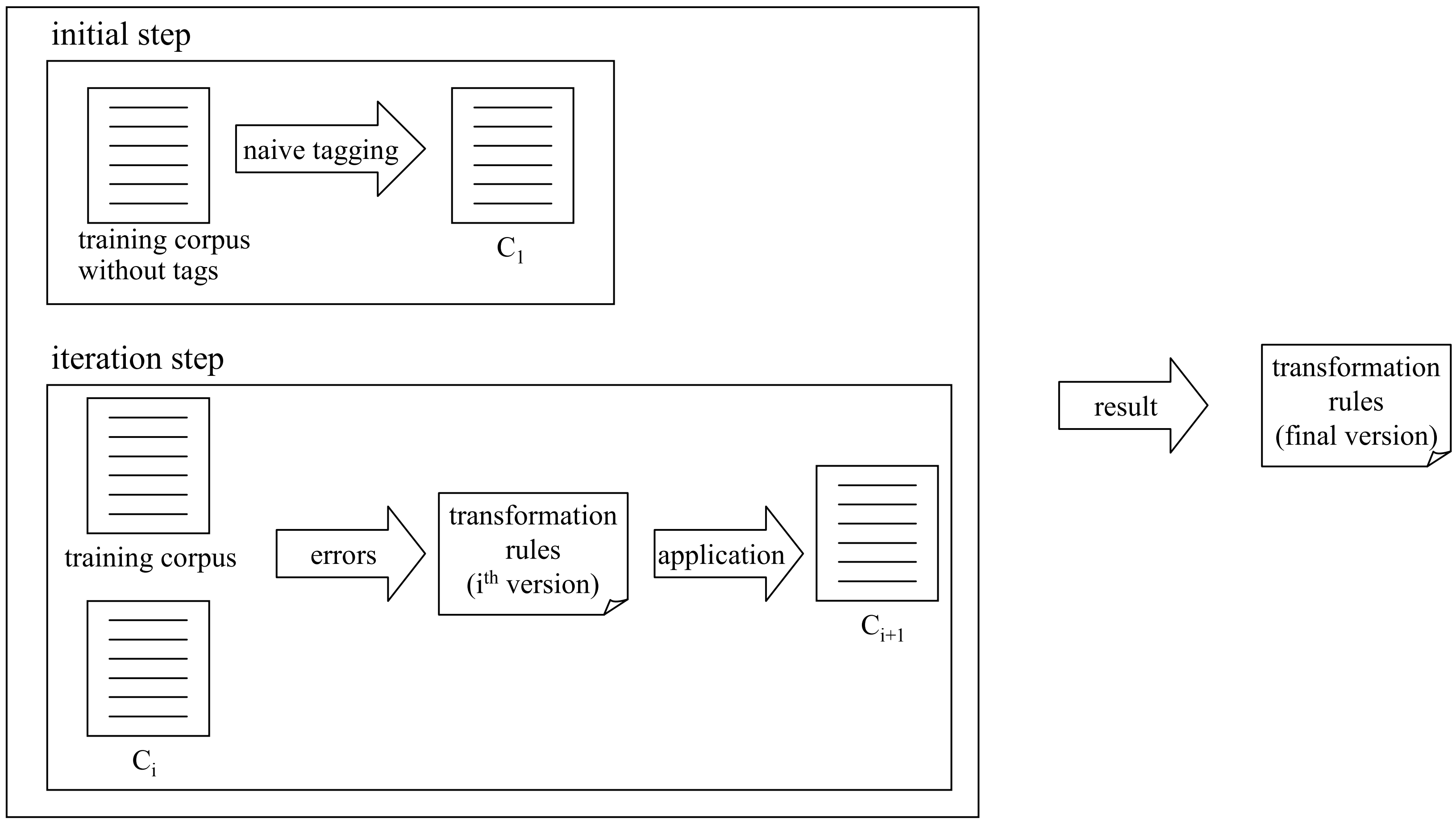


Main features of TnT:

- It uses trigrams of tags to compute transition probabilities
- Linear interpolation smoothing method
- Suffix analysis for unknown words
- Very fast

TBL. Transformation Based Learning

Transformation-based error-driven learning algorithm



Measures

$$\text{Precision} = \frac{\text{correctly recognized entities}}{\text{recognized entities}}$$

$$\text{Recall} = \frac{\text{correctly recognized entities}}{\text{actual entities}}$$

$$F_{\beta=1} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$$

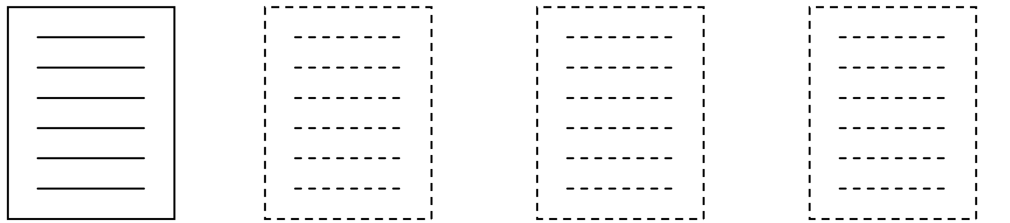
Baselines

Results obtained when TBL and TnT are trained with the original training corpus.

	Precision	Recall	$F_{(\beta=1)}$
TnT	84.39%	86.12%	85.25%
TBL	85.34%	89.66%	87.45%

Why do we need corpus transformation?

We use this technique to obtain different taggers using the same training corpus.



original corpus transformation 1 transformation 2 transformation 3

X 2 trainable taggers (TBL, TnT)

---

8 different taggers

Vocabulary reduction (I)

- Regular expressions are used to identify orthographical features of words (e.g. starting by capitals, containing digits,...)

- If a word matches a regular expression, it is substituted by the corresponding token.

- Words that appear frequently around or inside entities are excluded from this transformation (prepositions like *en* or *de*, articles like *la* or *el* , ...).

- The excluded words set is calculated automatically from the examples of the training corpus.

Vocabulary reduction (II)

El	O
presidente	O
del	O
COI	B
,	O
Juan	B
Antonio	I
Samaranch	I
,	O
...	...

El	O
presidente	O
del	O
_all_cap_	B
,	O
_starts_cap_	B
_starts_cap_	I
_starts_cap_	I
,	O
...	...

	Baseline:$F_{(\beta=1)}$	$F_{(\beta=1)}$
TnT-V	85.25%	86.63%
TBL-V	87.45%	88.10%

New tag set (I)

- B: that denotes the beginning of a named entity with more than one word.
- BE: that is assigned to a single-word named entity.
- I: that is assigned to words that are inside of a multiple-word named entity, except to the last word.
- E: assigned to the last word of a multiple-word named entity
- O: that preserves its original meaning, words outside a named entity.

New tag set (II)

El	O
presidente	O
del	O
COI	B
,	O
Juan	B
Antonio	I
Samaranch	I
,	O
...	...

El	O
presidente	O
del	O
COI	BE
,	O
Juan	B
Antonio	I
Samaranch	E
,	O
...	...

	Baseline:$F_{(\beta=1)}$	$F_{(\beta=1)}$
TnT-V	85.25%	86.83%
TBL-V	87.45%	87.91%

Adding POS information

Vocabulary reduction
+
POS information

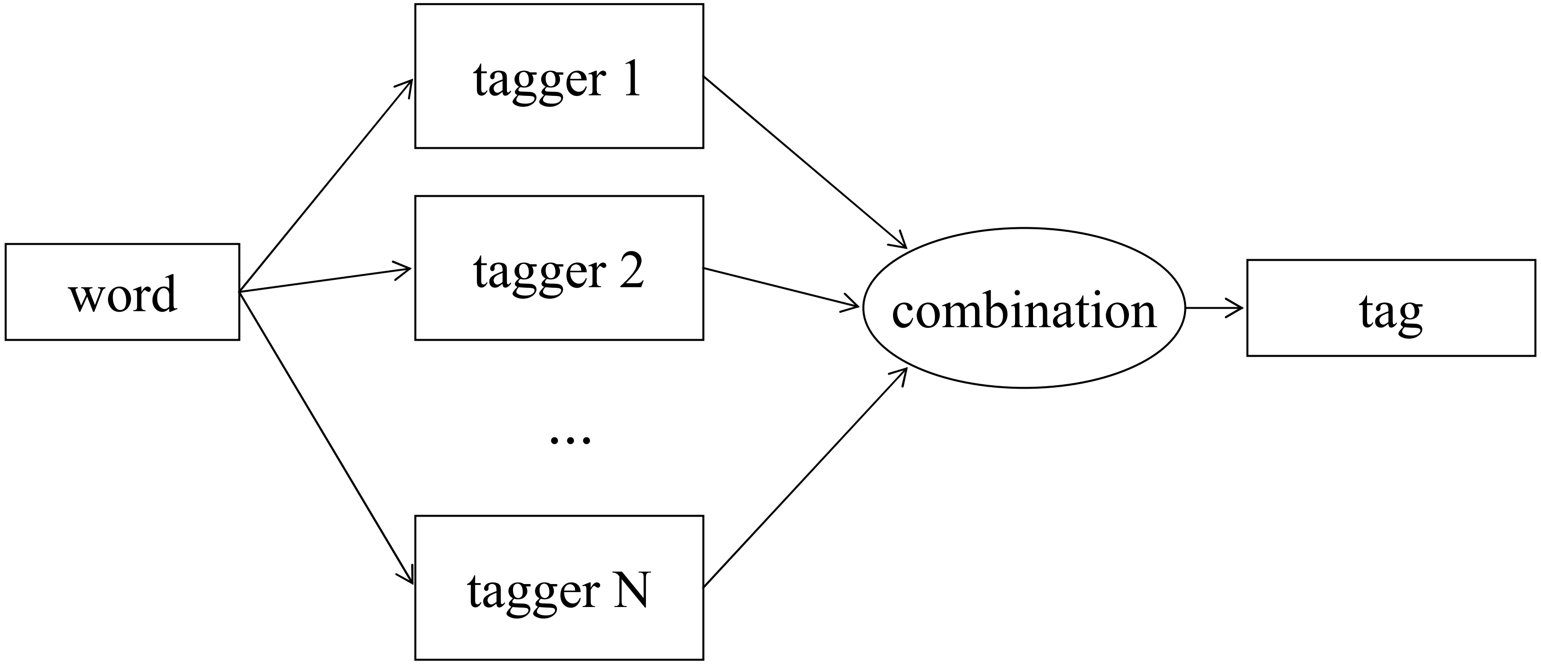
El	O
presidente	O
del	O
COI	B
,	O
Juan	B
Antonio	I
Samaranch	I
,	O
...	...

El_det_	O
presidente_noun_	O
del_prep_	O
_all_cap__noun_	B
,_punt_	O
_starts_cap__noun_	B
_starts_cap__noun_	I
_starts_cap__noun_	I
,	O
...	...

	Baseline:$F_{(\beta=1)}$	$F_{(\beta=1)}$
TnT-P	85.25%	86.69%
TBL-P	87.45%	89.22%

Combining different taggers

If you have to make an important decision, you usually consult several experts.



Voting and Stacking are common combination methods.

Voting

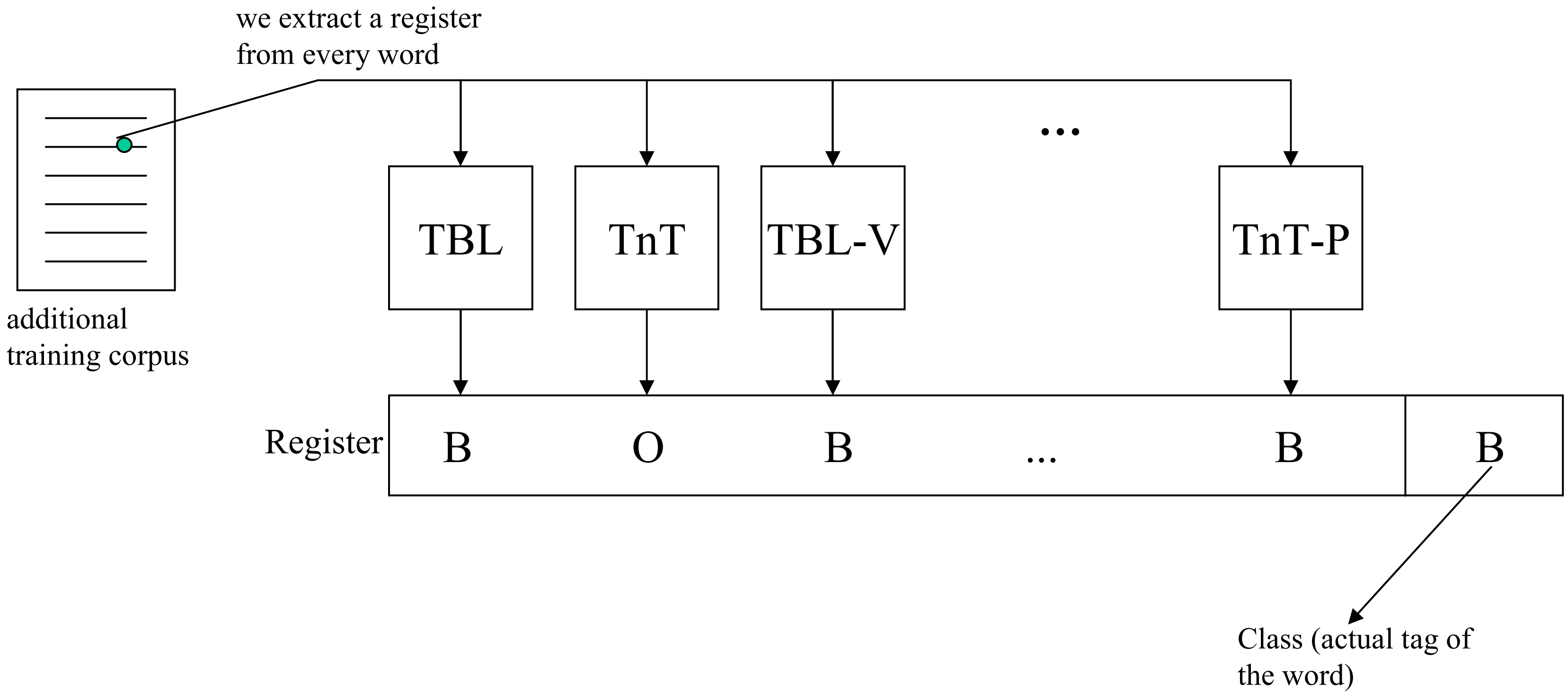
We have carried out two different experiments:

- Voting: the opinion of each model participant in the combination is equally important.

- Weighted voting: the vote of a model is weighted according to its performance in a previous evaluation.

Stacking: ML as a combination method

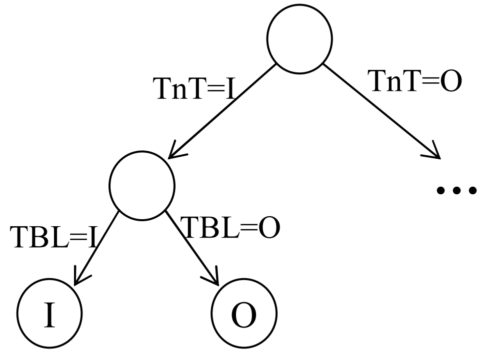
The use of automatic classification (machine learning) techniques for combining the results of the classifiers.



Stacking. Decision trees

TnT	**TBL**	**...**	**TBL-P**	**tag**
O	O	...	O	O
I	O	...	I	I
B	I	...	I	B
...	...	...	...	...
I	I	...	I	I

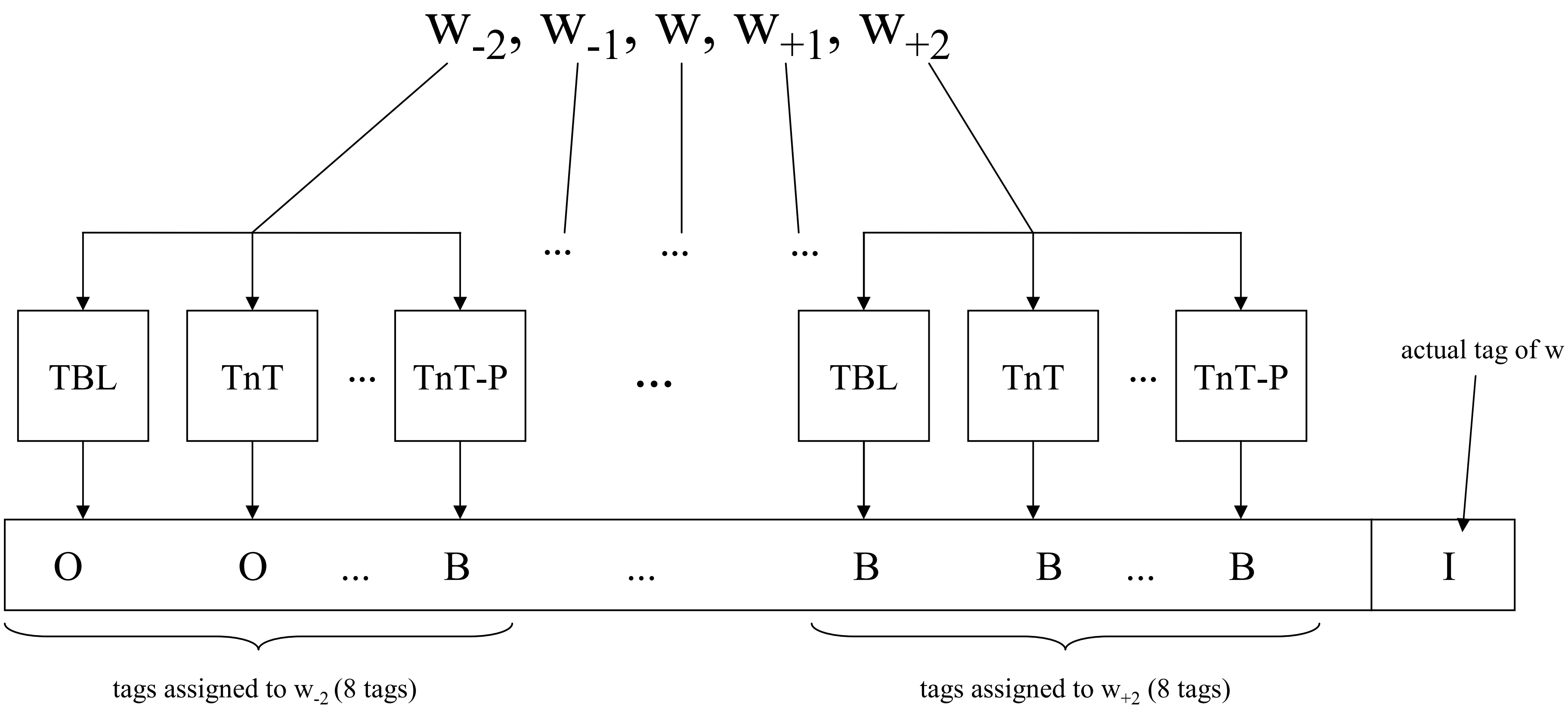
Training Database



Model

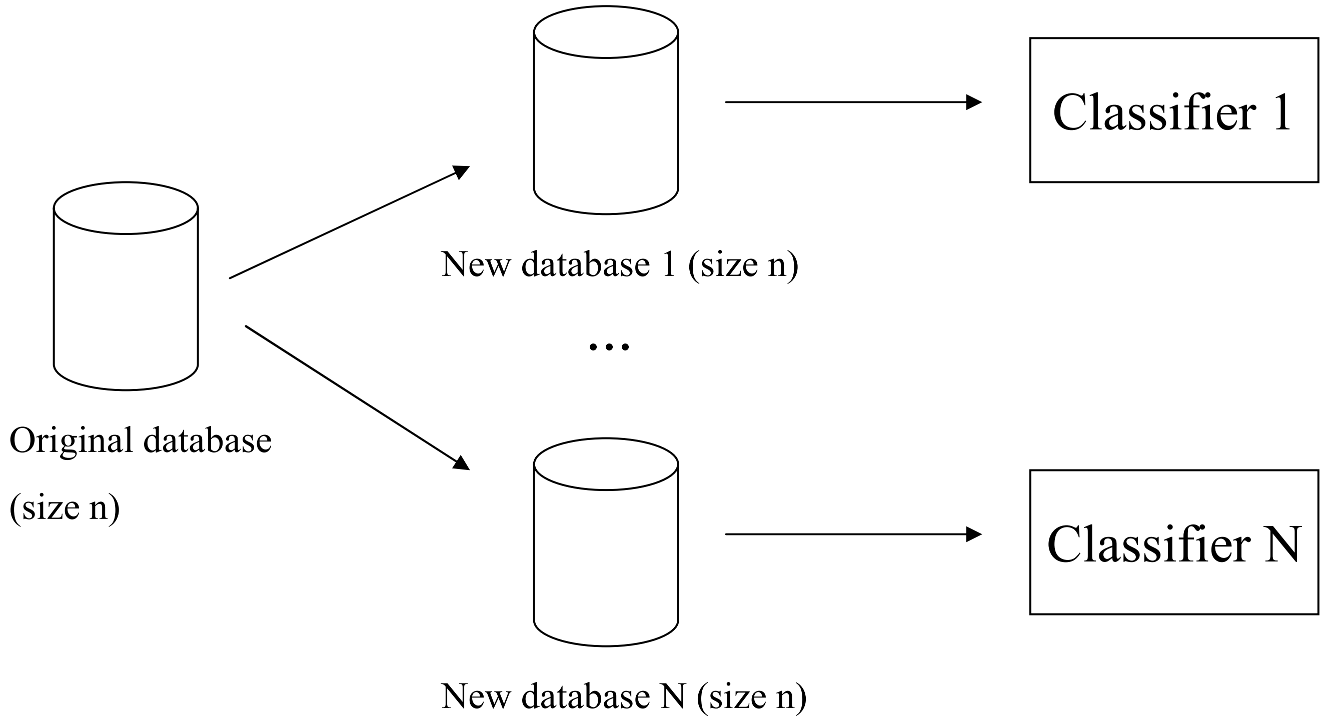
Stacking. Adding contextual information

We also include in the registers the tags assigned to the two previous words and the two following words.



Stacking. Bagging as learning method

Each new training database is obtained randomly sampling the original database. New databases can contain repeated examples.



Results

	$F_{(\beta=1)}$
Voting	89.97%
Weighted voting	90.02%
Stacking (decision trees)	89.72%
Stacking (contextual information)	90.48%
Stacking (contextual information + bagging)	90.90%

TBL baseline : 87.45%

TnT baseline : 85.25%

Conclusions

- All the transformations studied improve the results obtained with the original version of the corpus (reason: recognition is a simpler problem than extraction).
- System combination improves the results of individual taggers (reason: a wrong answer can be corrected with right answers of the rest of the taggers).
- Stacking is a better combination method than voting (reason: we can manage situations in which one tagger is right and the rest are wrong).

Future work

- Recognition of entities in specific domains or other NLP tasks.
- We can use properties of the domain to implement specific transformations.
- To apply these ideas in active learning or co-training techniques using this kind of system combination as agreement criterion.