Measuring the Relatedness between Documents in Comparable Corpora

Hernani Costa[a], Gloria Corpas Pastor[a] and Ruslan Mitkov[b]

{hercos,gcorpas}@uma.es, r.mitkov@wlv.ac.uk

[a]LEXYTRAD, University of Malaga, Spain
[b]RIILP, University of Wolverhampton, UK

November, 2015

Overview

- Comparable Corpora (CC)
  - automatic and assisted translation
  - language teaching
  - terminology
- Describing, comparing and evaluating CC
  - lack of standards

- This work aims at investigating the use of Distributional Similarity Measures (DSMs) as a tool to assess CC by
  - extracting
  - measuring
  - ranking

Motivation

- An inherent problem to those who deal with CC in a daily basis is the uncertainty about the data they are dealing with
  - tags like "casual speech transcripts" or "tourism specialised comparable corpus" are not enough to describe a corpus
- Most of the resources at our disposal are
  - built and shared without deep analysis of their content
  - used without knowing nothing about the relatedness quality of the corpus

Objectives

Investigate the use of textual DSMs in the context of CC

- automatically measure the relatedness between docs
- describe CC through the DSMs output scores
- analyse which features perform better
- rank docs by their degree of relatedness

Methodology

1) Data Preprocessing

- Sentence Detector and Tokeniser – OpenNLP[1]
- POS tagger and lemmatisation – TT4J[2]
- Stemming – Snowball[3]
- Stopword list[4]

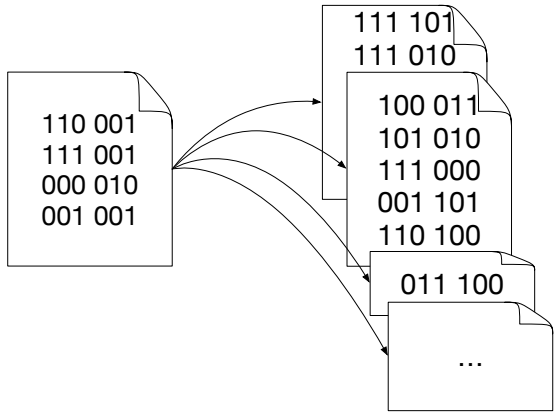
2) Identifying the list of common entities between docs

- Three co-occurrence matrices
  - common tokens, common lemmas and common stems

[1] https://opennlp.apache.org
[2] http://reckart.github.io/tt4j/
[3] http://snowball.tartarus.org
[4] https://github.com/hpcosta/stopwords

Methodology

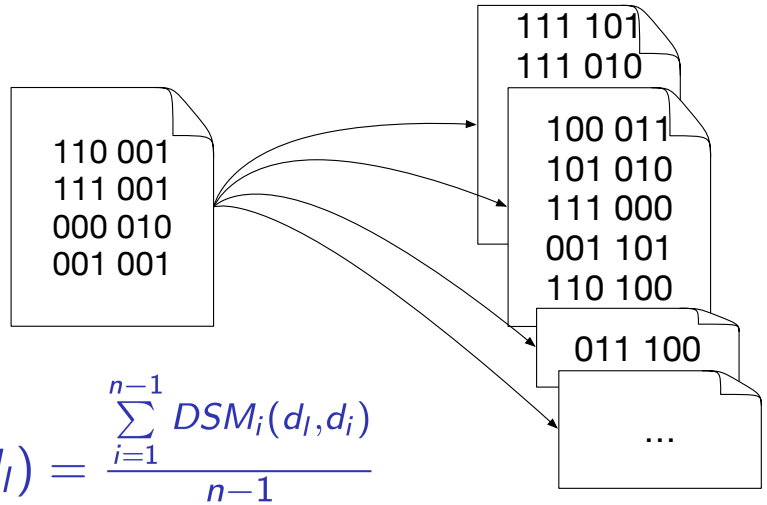
3) Computing the similarity between docs



- Input: list of common tokens, lemmas and stems
- $DSMs = \{DSM_{CE}, DSM_{SCC}, DSM_{\chi^2}\}$
  - CE: number of Common Entities
  - SCC: Spearman's Rank Correlation Coefficient
  - χ^2: Chi-Square

Methodology

4) Computing the doc final score



$$DSM(d_l) = \frac{\sum_{i=1}^{n-1} DSM_i(d_l, d_i)}{n-1}$$

where

- n: total number of docs
- $DSM_i(d_l, d_i)$: the resulted similarity score between the doc d_l with all the docs

5) Ranking docs

- descending order according to their DSMs scores

Corpora

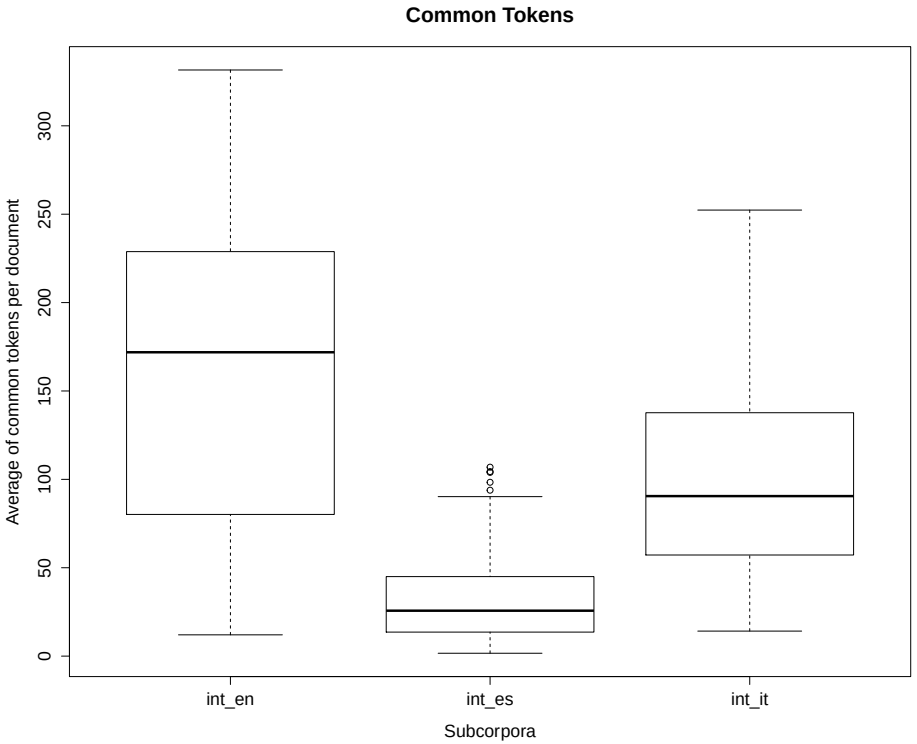
Statistical information about the various subcorpora

	nDocs	types	tokens	$\frac{types}{tokens}$
int_en	151	11,6k	496,2k	0.023
eur_en	30	3.4k	29,8k	0.116
int_es	224	13,2k	207,3k	0.063
eur_es	44	5,6k	43,5k	0.129
int_it	150	19,9k	386,2k	0.052
eur_it	30	4,7k	29,6k	0.159

- int_en, int_es and int_it: INTELITERM's docs in English, Spanish and Italian
- eur_en, eur_es and eur_it: docs randomly selected from the "one per day" Europarl v.7

INTELITERM corpus

Descriptive Statistics

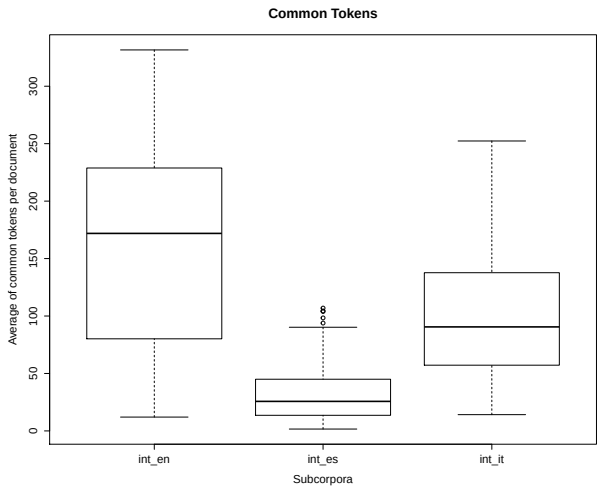


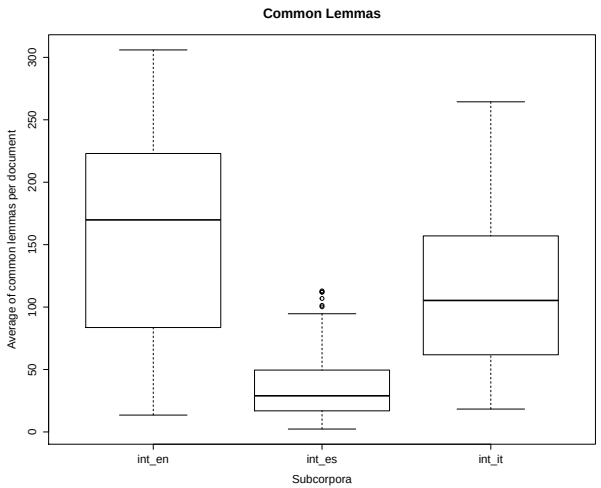
Average and standard deviation of common tokens scores between docs per subcorpus

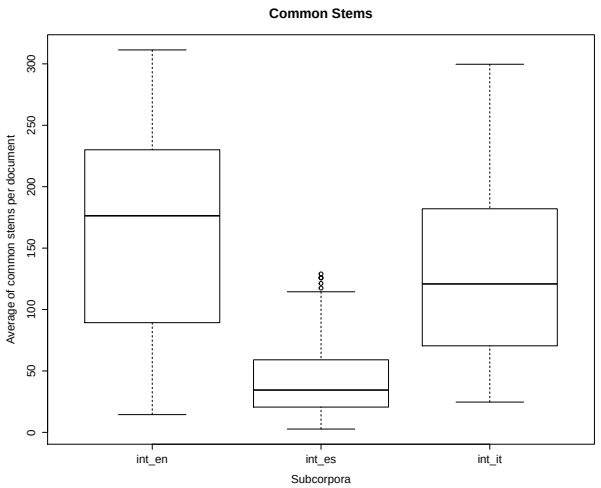
		NCT
int_en	av	163.70
	σ	83.87
int_es	av	31.97
	σ	23.48
int_it	av	101.08
	σ	55.71

INTELITERM corpus

General Findings



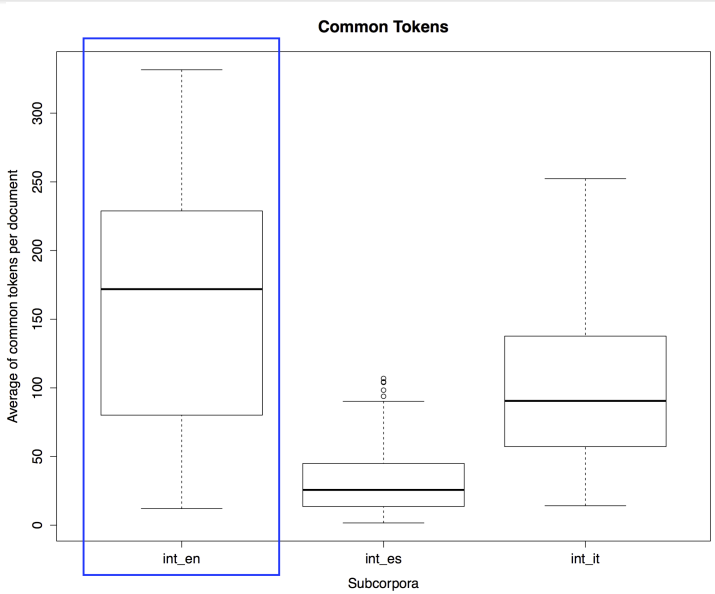




- scores for each subcorpus is roughly symmetric
  $\rightarrow$ data is normally distributed
- distributions between the features are quite similar
  $\rightarrow$ it is possible to achieve acceptable results only using tokens

INTELITERM corpus

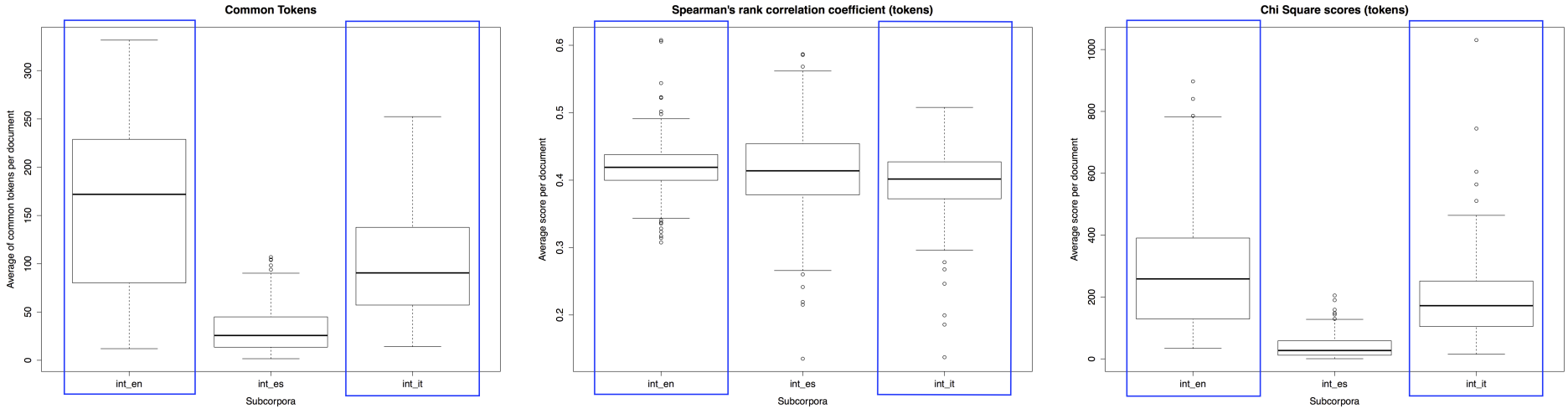
EN vs. ES & IT



- NCT per doc on average is higher + large IQR + long whiskers + skewed left
  → data is more spread + average of NCT per doc is more variable + wide type of docs (either highly or roughly correlated to the rest of the docs)
  → but, in general, docs have a high degree of relatedness between each other

INTELITERM corpus

EN & IT vs. ES



- From the statistical and theoretical evidences
  $\rightarrow$ NCT: high + SCC: high average scores +
  χ^2: long whisker outside the upper quartile
  $\rightarrow$ EN and IT subcorpora look like they assemble highly correlated docs
  $\rightarrow$ docs have a high degree of relatedness between each other
- Is int_es composed by low related docs?

Measuring DSMs Performance

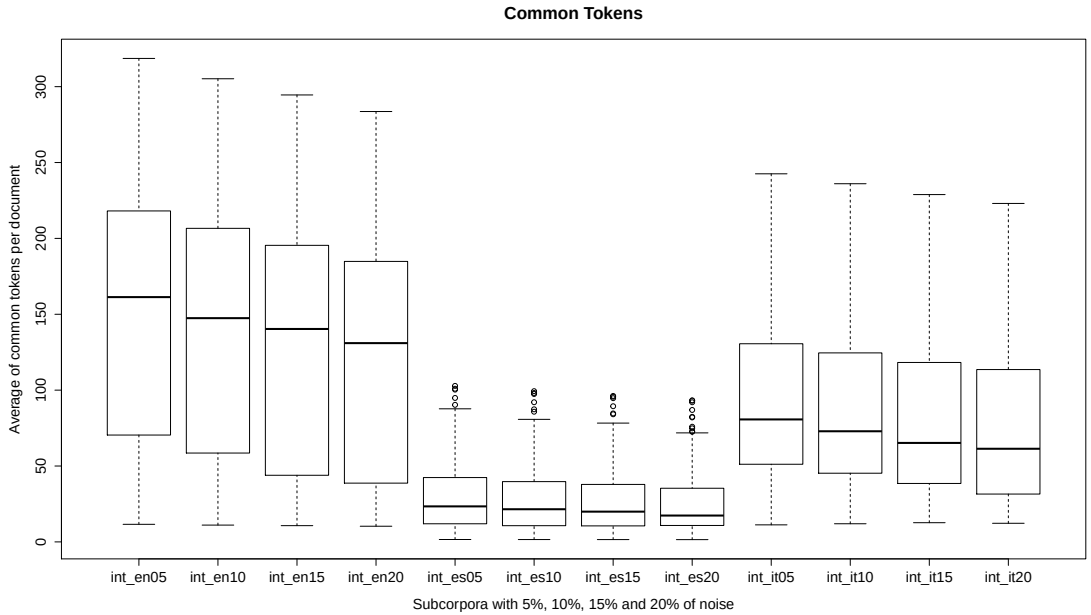
Goal

- How do the DSMs perform the task of filtering out docs with a low level of relatedness?

- Set-up
  - inject different sets of out-of-domain docs, randomly selected from the Europarl corpus to the INTELITERM subcorpora

Measuring DSMs Performance

Average scores between docs when injecting 5%, 10%, 15% and 20% of noise



- the more noisy docs are injected, the lower is the NCT
- Next step: rank docs in a descending order according to their DSMs scores and evaluate their precision

Measuring DSMs Performance

DSMs precision when injecting different amounts of noise to the various subcorpora

SubC	Noise	NCT	SCC	χ^2
int_en	*5%*	0.89	0.22	**1.00**
	10%	0.73	0.33	**1.00**
	15%	0.73	0.36	0.95
	20%	**0.80**	0.37	**0.90**
int_es	*5%*	0.00	0.00	0.38
	10%	0.07	0.07	0.20
	15%	0.09	0.09	0.17
	20%	0.14	0.18	0.23
int_it	*5%*	0.88	0.13	0.88
	10%	0.82	0.06	0.82
	15%	0.74	0.09	0.83
	20%	**0.73**	0.13	0.87

- none of the DSMs got acceptable results for Spanish
  - due to the pre-existing low level of relatedness
- promising results for English and Italian
  - NCT and χ^2 performed well

Summary

From the statistical and theoretical evidences

- int_en and int_it
  - assemble highly correlated docs
- int_es
  - scarceness of evidences only allow was to not reject the idea that this subcorpus is composed of similar docs
- NCT & χ^2
  - suitable for the task of filtering out low related docs with a high precision degree

Conclusion

- DSMs can be used to describe and measure the relatedness between docs in specialised CC
  - three different input features were used (lists of common tokens, lemmas and stems)
  - for the data in hand, these features had similar performance for all the tested DSMs
- INTELITERM corpus seems to be composed of highly correlated docs
  - high number of CE and positive average SCC and χ^2 scores

Current Work

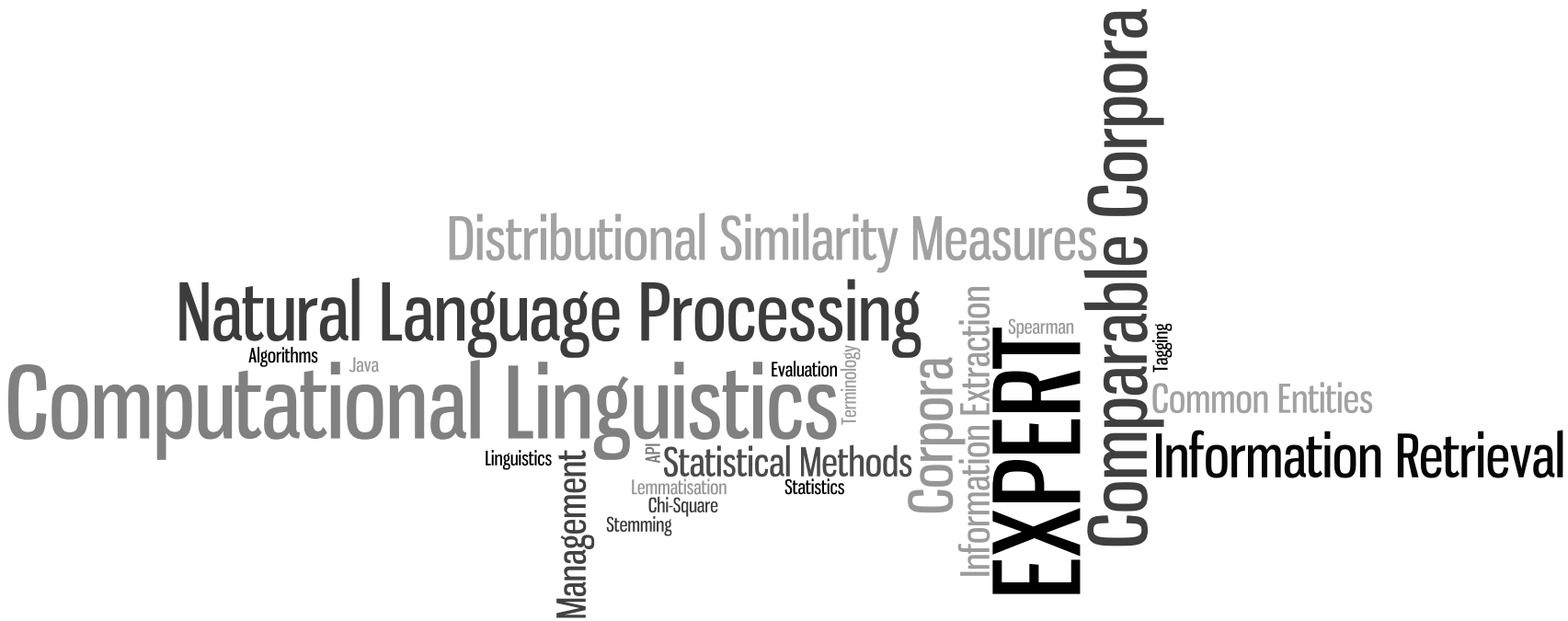
- Perform more experiments with DSMs
  - use other languages
  - evaluate other DSMs (e.g. Jaccard, Lin and Cosine)
  - compare corpora manual with semi-automatic compiled

$\rightarrow$ Using this approach to automatically filter out docs with a low level of relatedness

$\rightarrow$ will improve the precision of terminology extraction

Acknowledgements

Hernani Costa is supported by the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement n° 317471. Also, the research reported in this work has been partially carried out in the framework of the Educational Innovation Project TRADICOR (PIE 13-054, 2014-2015); the R&D project INTELITERM (ref. n° FFI2012-38881, 2012-2015); the R&D Project for Excellence TERMITUR (ref. n° HUM2754, 2014-2017); and the LATEST project (ref. 327197-FP7-PEOPLE-2012-IEF).


Distributional Similarity Measures
Natural Language Processing
Algorithms
Java
Computational Linguistics
Evaluation
Terminology
Linguistics
API
Statistical Methods
Statistics
Lemmatisation
Chi-Square
Stemming
Management
Corpora
Information Extraction
EXPERT
Spearman
Comparable Corpora
Tagging
Common Entities
Information Retrieval