4th Italian Information Retrieval Workshop
January 16-17, 2012 Pisa, Italy

Distributional Semantics for Answer Re-ranking in Question Answering*

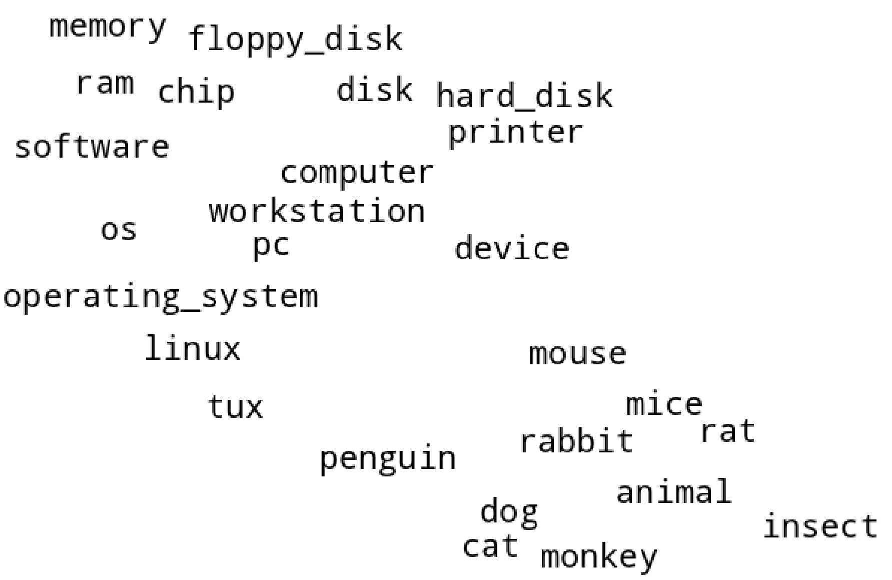
Piero Molino, Pierpaolo Basile, Annalina Caputo, Pasquale Lops, Giovanni Semeraro
{pierpaolo.basile@uniba.it}
Dept. of Computer Science, University of Bari Aldo Moro

* This paper summarizes the main results already published in Molino, P., Basile, P.,Caputo, A., Lops, P., Semeraro, G.: Exploiting Distributional Semantic Models in Question Answering. In: 6th IEEE-ICSC 2012, Palermo, Italy.

Background

You shall know a word by the company it keeps!

Meaning of a word is determined by its usage



Background

Distributional Semantic Models (DSMs)

- represent words as points in a geometric space
- do not require specific text operations
- corpus/language independent

Widely used in IR and Computational Linguistic

- semantic text similarity, synonyms detection, query expansion, …

Never been used for candidate answers re-ranking

Our idea

- Using DSMs into a QA system for candidate answers re-ranking
  - QuestionCube: a framework for QA
- Exploit DSMs
  - compare semantic similarity between user's question and candidate answers

Wikiedi
alpha

www.wikiedi.it

DEMO

Distributional Semantic Models

Term-term co-occurrence matrix: each cell contains the co-occurrences between two terms within a prefixed distance

	dog	cat	computer	animal	mouse
dog	0	4	0	2	1
cat	4	0	0	3	5
computer	0	0	0	0	3
animal	2	3	0	0	2
mouse	1	5	3	2	0

…Distributional Semantic Models

TTM: Term-Term co-occurrence Matrix

Latent Semantic Analysis (LSA): relies on the Singular Value Decomposition (SVD) of the co-occurrence matrix

Random Indexing (RI): based on the Random Projection

Latent Semantic Analysis over Random Indexing (LSARI)

Random Indexing

1. Generate and assign a context vector to each context element (e.g. document, passage, term, …)
2. Term vector is the sum of the context vectors in which the term occurs
   – sometimes the context vector could be boosted by a score

- RI is a locality-sensitive hashing method which approximate the cosine distance between vectors

Latent Semantic Analysis over Random Indexing

1. Reduce the dimension of the co-occurrences matrix using RI
2. Perform LSA over RI (LSARI)
   – reduction of LSA computation time: RI matrix contains less dimensions than co-occurrences matrix

QuestionCube framework…

Natural Language pipeline for Italian and English

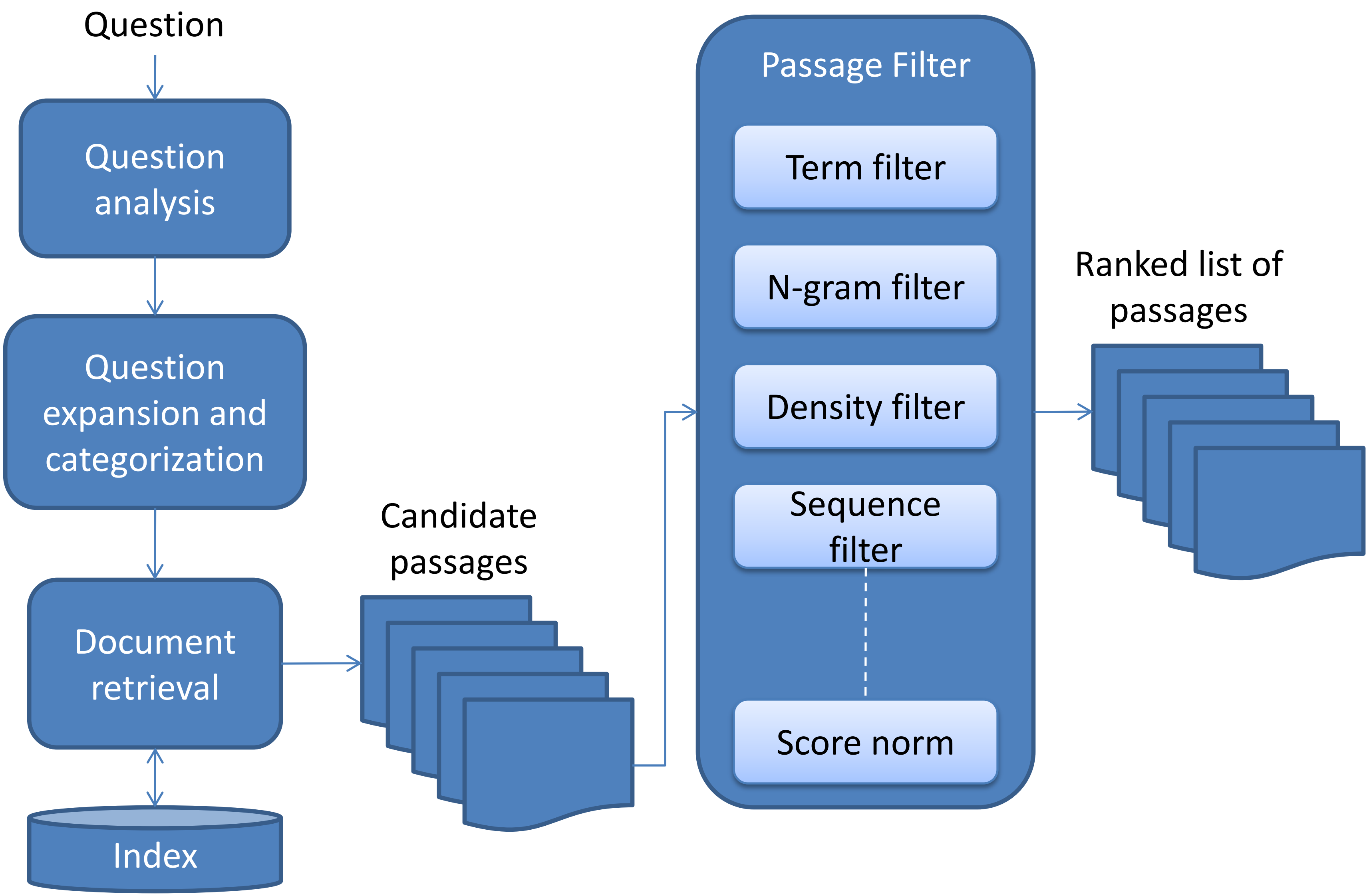
- PoS-tagging, lemmatization, Name Entity Recognition, Phrase Chunking, Dependency Parsing, Word Sense Disambiguation (WSD)

IR model based on classical VSM or BM25

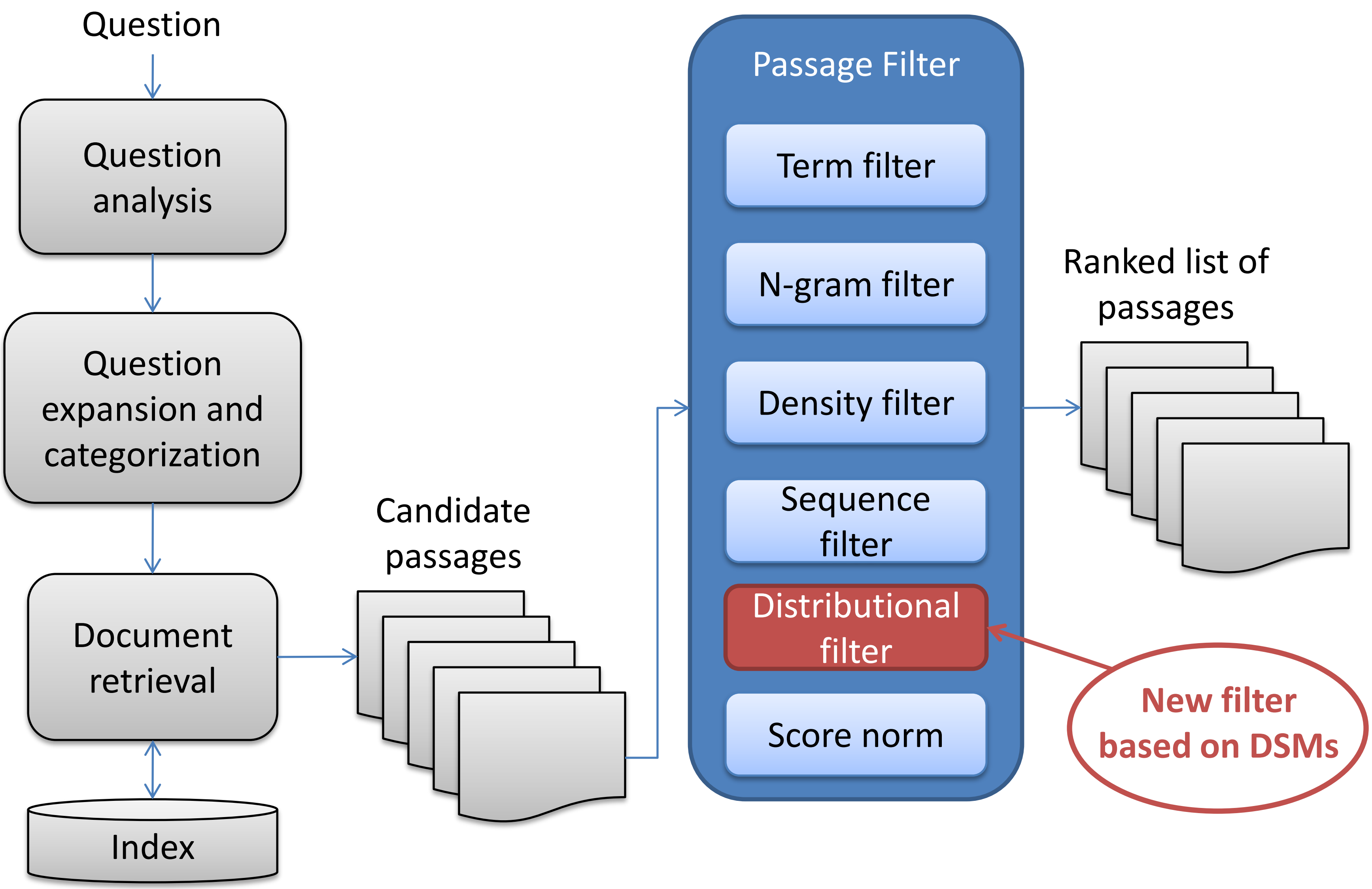
- several query expansion strategies

Passage retrieval

…QuestionCube framework



Integration



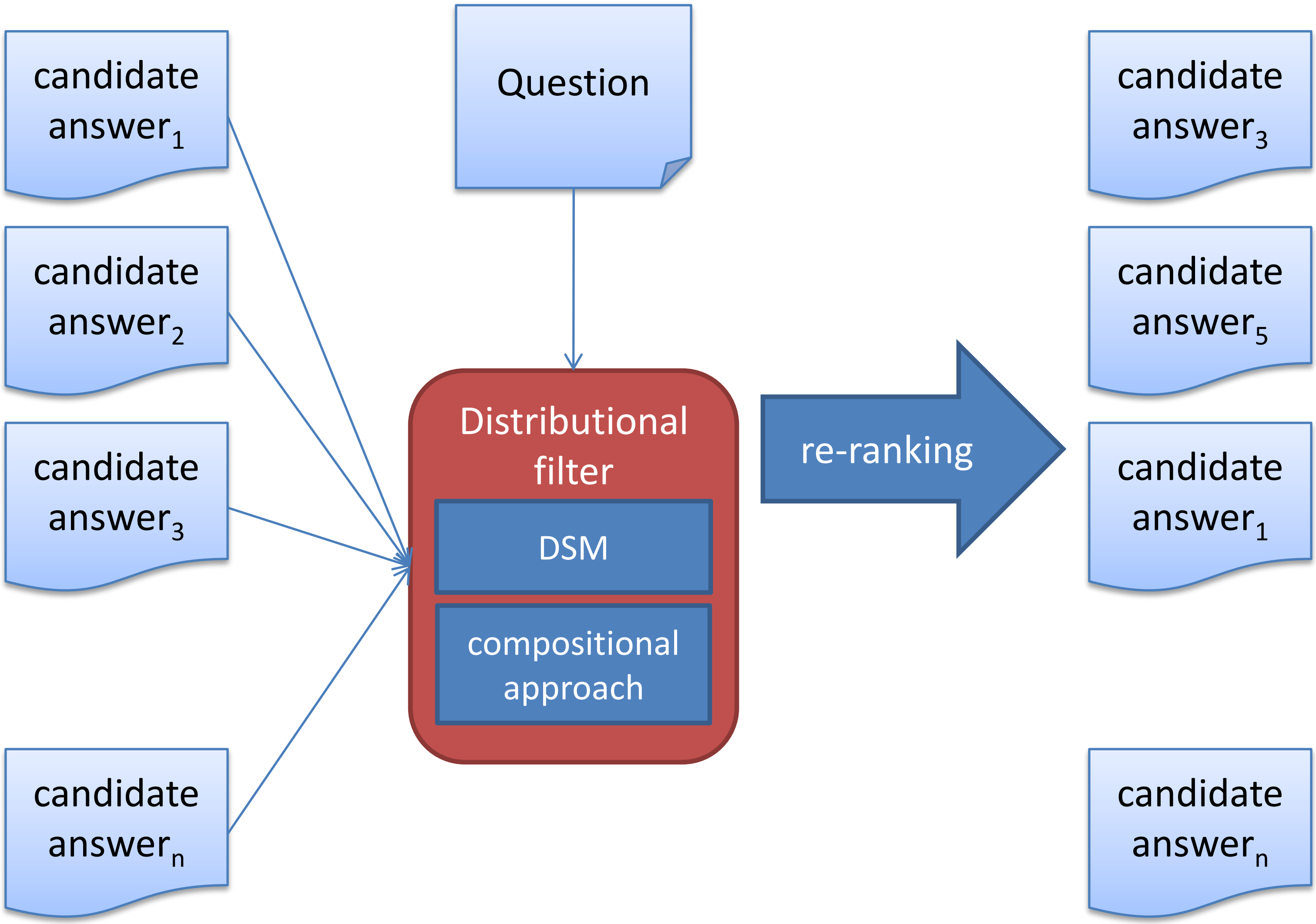
Distributional filter…

Compute the similarity between the user's question and each candidate answer

Both user's question and candidate answer are represented by more than one term

- a method to compose words is necessary

…Distributional filter…



…Distributional filter (compositional approach)

Addition (+): pointwise sum of components

- represent complex structures by summing words which compose them

Given $q = q_1\ q_2 \dots q_n$ and $a = a_1\ a_2 \dots a_m$

- question: $\vec{q} = \vec{q}_1 + \vec{q}_2 + \ldots + \vec{q}_n$
- candidate answer: $\vec{a} = \vec{a}_1 + \vec{a}_2 + \ldots + \vec{a}_m$
- similarity $sim(q, a) = \dfrac{\vec{q} \cdot \vec{a}}{\|\vec{q}\| \|\vec{a}\|}$

Evaluation…

Dataset: 2010 CLEF QA Competition

- 10,700 documents
  - from European Union legislation and European Parliament transcriptions
- 200 questions
- Italian and English

DSMs

- 1,000 vector dimension (TTM/LSA/RI/RILSA)
- 50,000 most frequent words
- co-occurrence distance: 4

...Evaluation

1. Effectiveness of DSMs in QuestionCube
2. Comparison between the several DSMs adopted

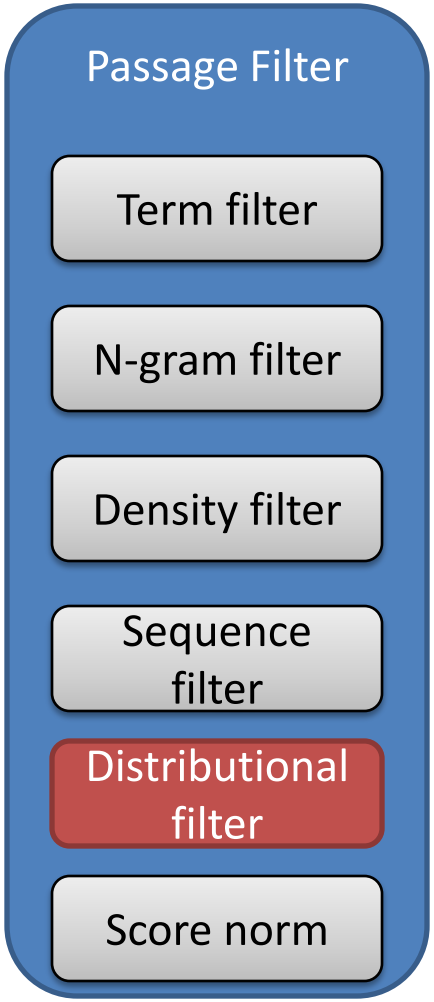
Metrics

- a@*n*: accuracy taking into account only the first *n* answers
- MRR based on the rank of the correct answer

$$\frac{\sum_{i=1}^{N} \frac{1}{rank_i}}{N}$$

Evaluation (alone scenario)

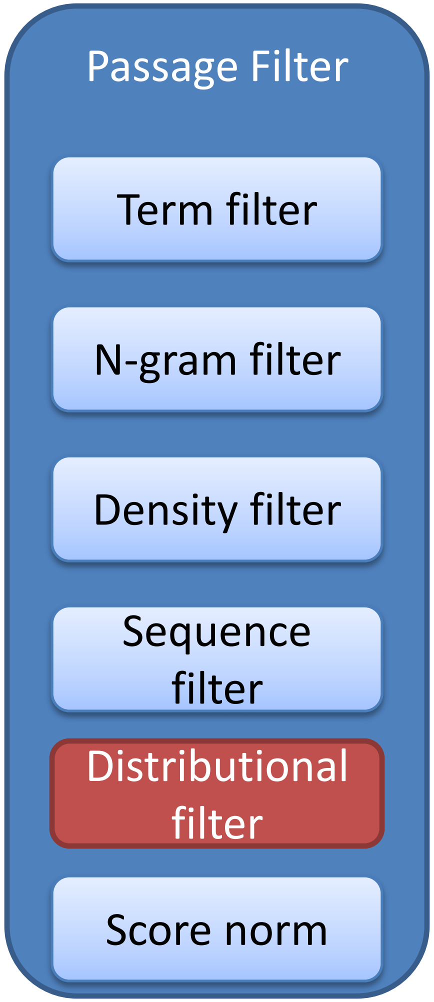
Only distributional filter is adopted: no other filters in the pipeline



Evaluation (combined)

Distributional filter is combined with the other filters

- using CombSum function
- *baseline*: distributional filter is removed



Results (English)…

	Run	a@1	MRR
alone	TTM	.060	.107
	RI	.180	.267
	LSA	.205	.300
	LSARI	.190	.295
combined	*baseline**	*.445*	*.549*
	TTM	.535	.614[1]
	RI	.550	**.637**[1,2]
	LSA	.560	**.637**[1]
	LSARI	.555	.634[1]

[1]significat wrt. baseline
[2]significat wrt. TTM
*without distributional filter

…Results (Italian)

	Run	a@1	MRR
alone	TTM	.060	.097
	RI	.175	.241
	LSA	.155	.229
	LSARI	.180	.254
combined	*baseline**	*.365*	*.441*
	TTM	.405	.539[1]
	RI	.465	.555[1]
	LSA	.470	.551[1]
	LSARI	.480	**.557**[1,2]

[1]significat wrt. baseline
[2]significat wrt. TTM
*without distributional filter

Final remarks…

Alone: all the proposed DSMs perform better than the TTM

- in particular LSA and LSARI

Combined: all the combinations are able to overcome the baseline

- English +16% (RI/LSA), Italian +26% (LSARI)

No remarkable difference in performance between LSA and LSARI

…Final remarks

Distributional filter in the alone scenario shows an higher improvement than in the combination scenario

Future work

- –find more effective way to combine filters (learning to rank approach)
- –Exploit more complex semantic compositional strategies

Thank you for your attention!

Questions?

<pierpaolo.basile@uniba.it>

BACKUP SLIDES

Context Vector

0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 -1 0 1 0 0 0 0 1 0 0 0 0 -1

- sparse
- high dimensional
- ternary {-1, 0, +1}
- small number of randomly distributed non-zero elements

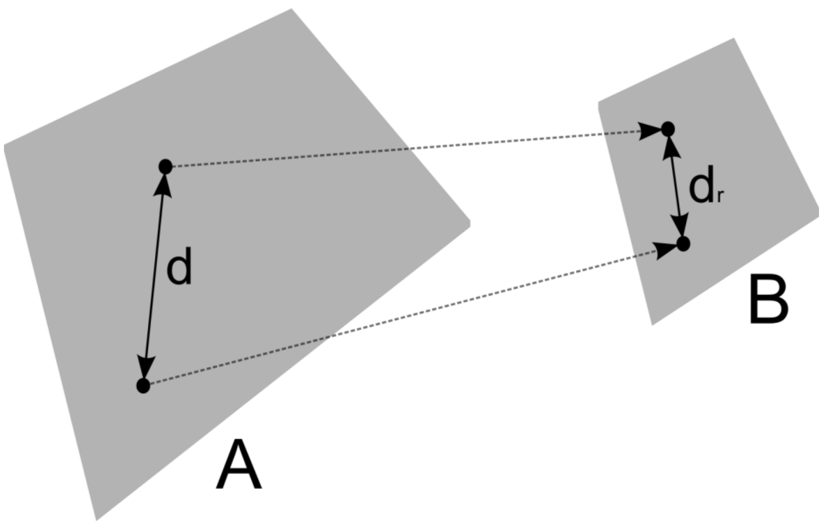
Random Indexing (example)

John eats a red <u>apple</u>

CV_{john} -> (0, 0, 0, 0, 0, 0, 1, 0, -1, 0)
CV_{eat} -> (1, 0, 0, 0, -1, 0, 0 ,0 ,0 ,0)
CV_{red} -> (0, 0, 0, 1, 0, 0, 0, -1, 0, 0)

TV_{apple}= CV_{john}+ CV_{eat}+ CV_{red}=(1, 0, 0, 1, -1, 0, 1, -1, -1, 0)

Random Indexing (formal)



$$B^{n,k} = A^{n,m} R^{m,k} \quad k << m$$

B nearly preserves the distance between points (Johnson-Lindenstrauss lemma)

$$d_r = c \times d$$

RI is a locality-sensitive hashing method which approximate the cosine distance between vectors