Semantic Models for Answer Re-ranking in Question Answering

Piero Molino - piero.molino@uniba.it
Università degli Studi di Bari Aldo Moro

28/7/2013 - SIGIR 2013

Question Answering (QA)

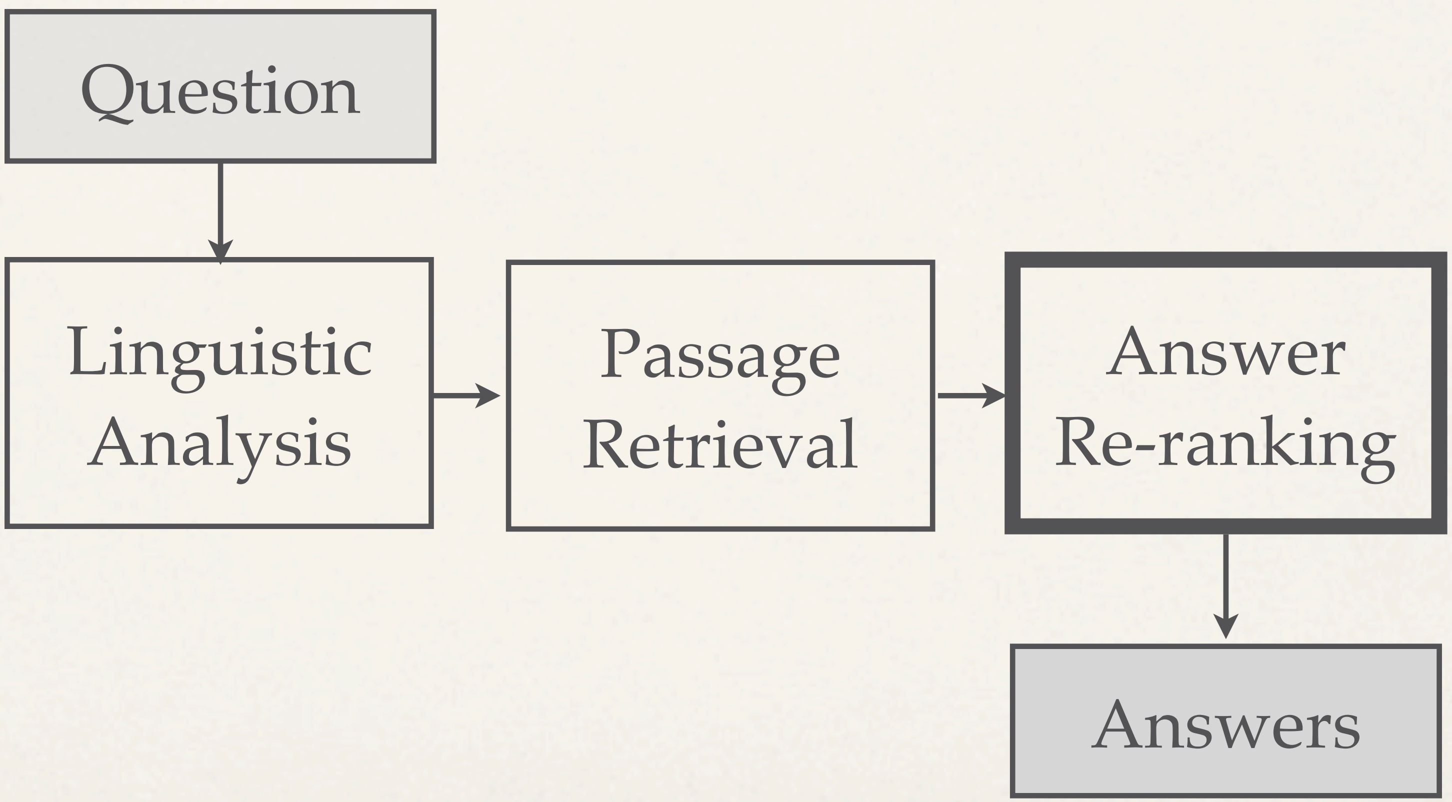
- **Query** = Natural Language Question
- **Result** = Exact Answer or Short Passage

- Who’s the adoptive son of Julius Cesar?
- Here we see Brutus, the adoptive son of Julius Cesar, hitting him with a dagger

Non-factoid QA

- Factoid
  - Who, Where, When
  - Answers are Named Entities, dates or numbers
  - Needs **structured data** or **extraction** from unstructured data
- Non-factoid
  - Causation, manner, reason
  - Answers are sentences or paragraphs
  - Needs **NLP** for question-answer similarity

General Architecture



Learning to Rank (MLR)

- **Learn** the Ranking Function from Question-Answer
- Represent Question-Answer pair as a datapoint with
  - **Question specific** and **Answer spacific** features (lenght, category, type of origin document, ...)
  - **Question-Answer features** (different similarity measures, TFIDF, BM25, N-gram overlap, Machine Translation, structural similarity, ...)

Semantic Models

- Exploit **latent** or **explicit concepts** rather than words
- Widely used in IR and Computational Linguistic for semantic text similarity, synonyms detection, query expansion, topic identification, ...
- Latent Semantic Analysis, Random Indexing, Latent Dirichlet Allocation, Non-negative Matrix Factorization, Explicit Semantic Analysis

Research Questions

- Are additional semantic features useful for answer re-ranking?
- Which of them is more effective and under which circumstances?
- Do semantic features bring information that is not present in the bag-of-words and structured features?

Work Done

- Implement a QA System with NLP pipeline and MLR
- Add semantic features from Distributional Semantic Models (LSA and RI)
- Perform a preliminary experiment with a subset of features
- Add more similarity, linguistic and semantic features
- Experiment different MLR algorithms on different dataset

Distributional Semantics

- The meaning of a word is determined by its **usage**

A bottle of **Tesgüino** is on the table
Everyone likes **Tesgüino**
Tesgüino makes you drunk
We make **Tesgüino** out of corn

- It is a corn beer

Distributional Semantic Models

- Represent words as points in a geometric space
- **Do not require** specific text operations (corpus/ language independent)
- Widely used in IR and Computational Linguistic
- Never been used for answer re-ranking

Objective

- Semantic similarity between Question and Answer
- Computed with Distributional Semantic Models
- Used as re-rank feature

- Q: Which beverages contain alcohol?
- A: Tesgüino makes you drunk

Co-occurrence Matrix

- Term-term co-occurrence matrix: contains the co-occurrences between 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

Approximations

- **TTM**: Term-Term co-occurrence Matrix
- **Latent Semantic Analysis** (LSA): TSVD of the co-occurrence matrix
- **Random Indexing** (RI): based on the Random Projection
- **Latent Semantic Analysis over Random Indexing** (LSARI)

Random Indexing

- RI is a locality-sensitive hashing method which approximate the cosine distance between vectors
- **Generate** and **assign** a Context Vector to each context element (e.g. document, passage, term, …) with K random values in {-1, 0, +1}
- Term Vector is the **sum** of the Context Vectors of all contexts in which the term **occurs**

Random Indexing

Dataset:

I drink **Tesgüino**

You drink **Tesgüino** beer

Context Vectors

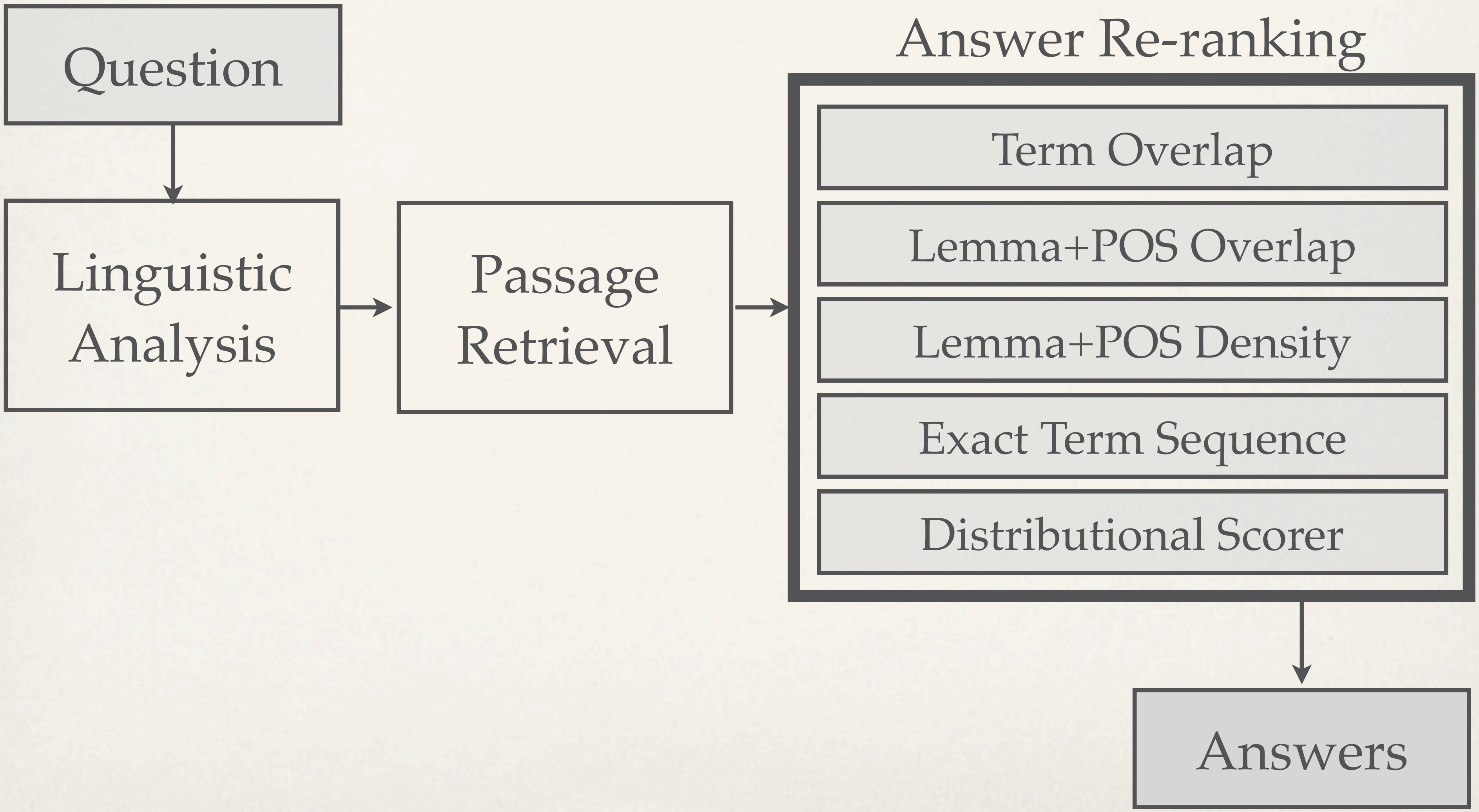
i	1	0	0	0	0	-1	0
drink	0	0	1	0	0	0	0
tesgüino	0	1	0	0	0	0	0
you	0	-1	0	0	0	0	1
beer	-1	0	0	0	1	0	0

Term Vector for **Tesgüino**

$1 \cdot cv_{i} + 2 \cdot cv_{drink} +$
$1 \cdot cv_{you} + 1 \cdot cv_{beer}$

tesgüino	0	-1	2	0	1	-1	1

Distributional Scorer



Compositionality

- We need a method to represent question and answers, as they are **composed** by more than one term
- **Addition** (+): sum of all the vectors of the terms in the question or answer
- Compute the **cosine similarity** between the summed vectors
- Other operators can be used (product, max, min, convolution, ...)

Evaluation

- Dataset: 2010 CLEF QA Competition
  - **10.700 documents** from European Union legislation and European Parliament transcriptions
  - **200 questions** in English and Italian
- DSMs
  - **1000** vector dimension (TTM/LSA/RI/LSARI)
  - **50.000** most frequent words
  - Co-occurrence distance: **4**

Objective and Metrics

- **Effectiveness** of DSMs for the task
- **Comparison** between the several DSMs adopted
- Metrics
  - **a@n**: accuracy taking into account only the first n answers
  - **MRR**: average of the inverse rank of the first correct answer

Scenarios

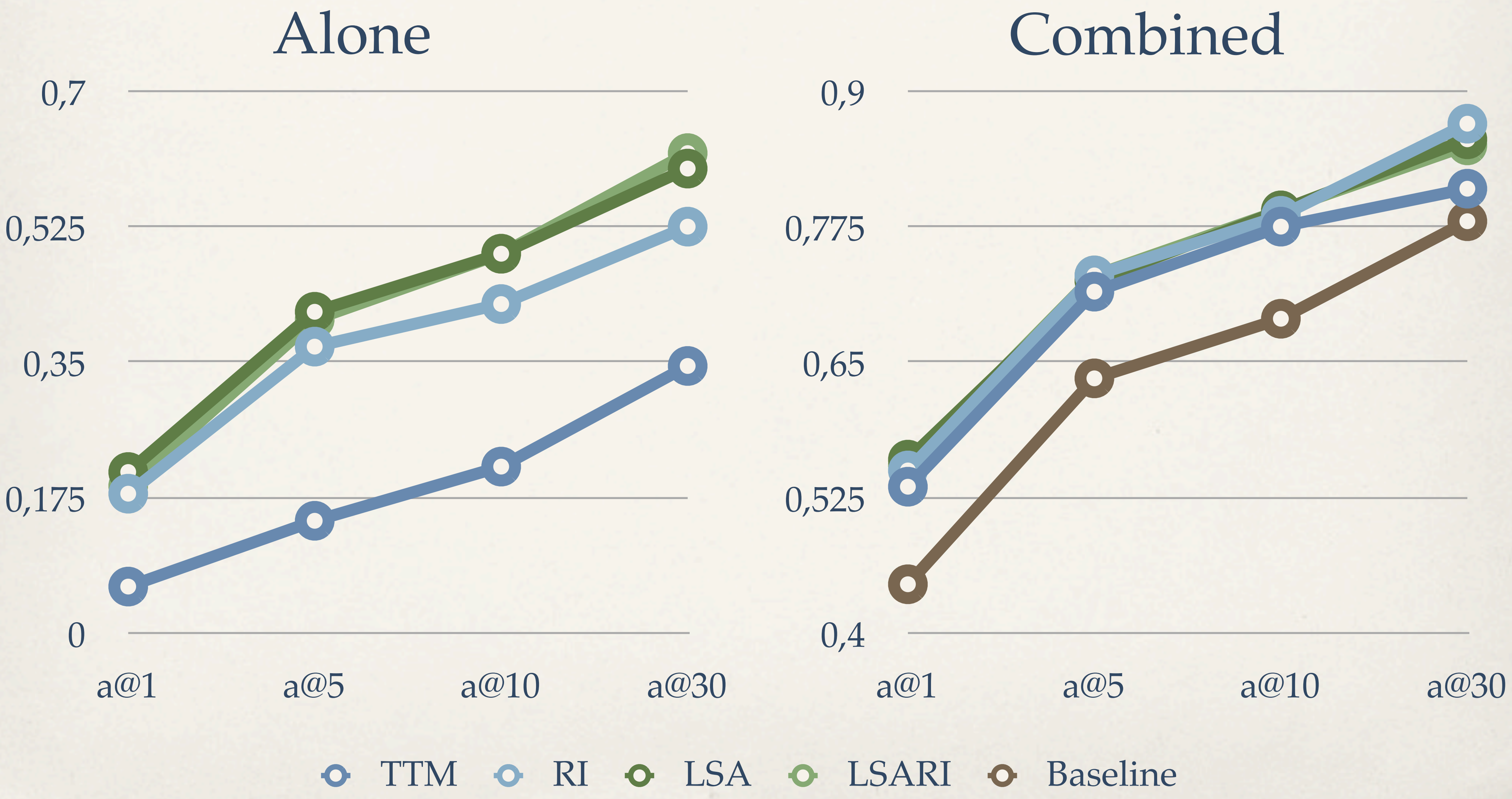
- Alone
- **Only** the Distributional scorer is adopted, no other scorers in the pipeline

Term Overlap
Lemma+POS Overlap
Lemma+POS Density
Exact Term Sequence
Distributional Scorer

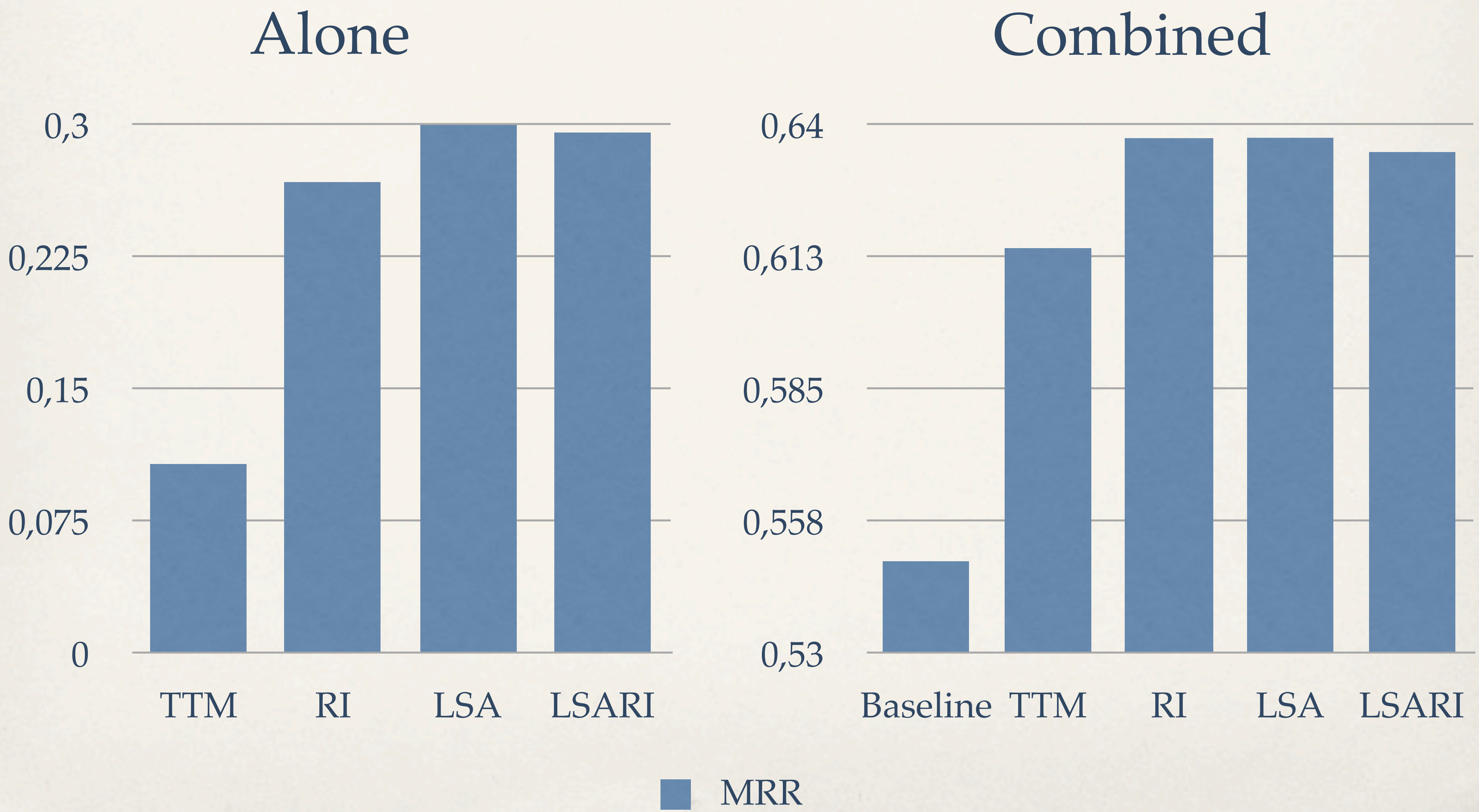
- Combined
- Distributional scorer **and** others with **CombSum**
- Baseline: distributional filter is **removed**

Term Overlap
Lemma+POS Overlap
Lemma+POS Density
Exact Term Sequence
Distributional Scorer

Results (English) a@n



Results (English) MRR



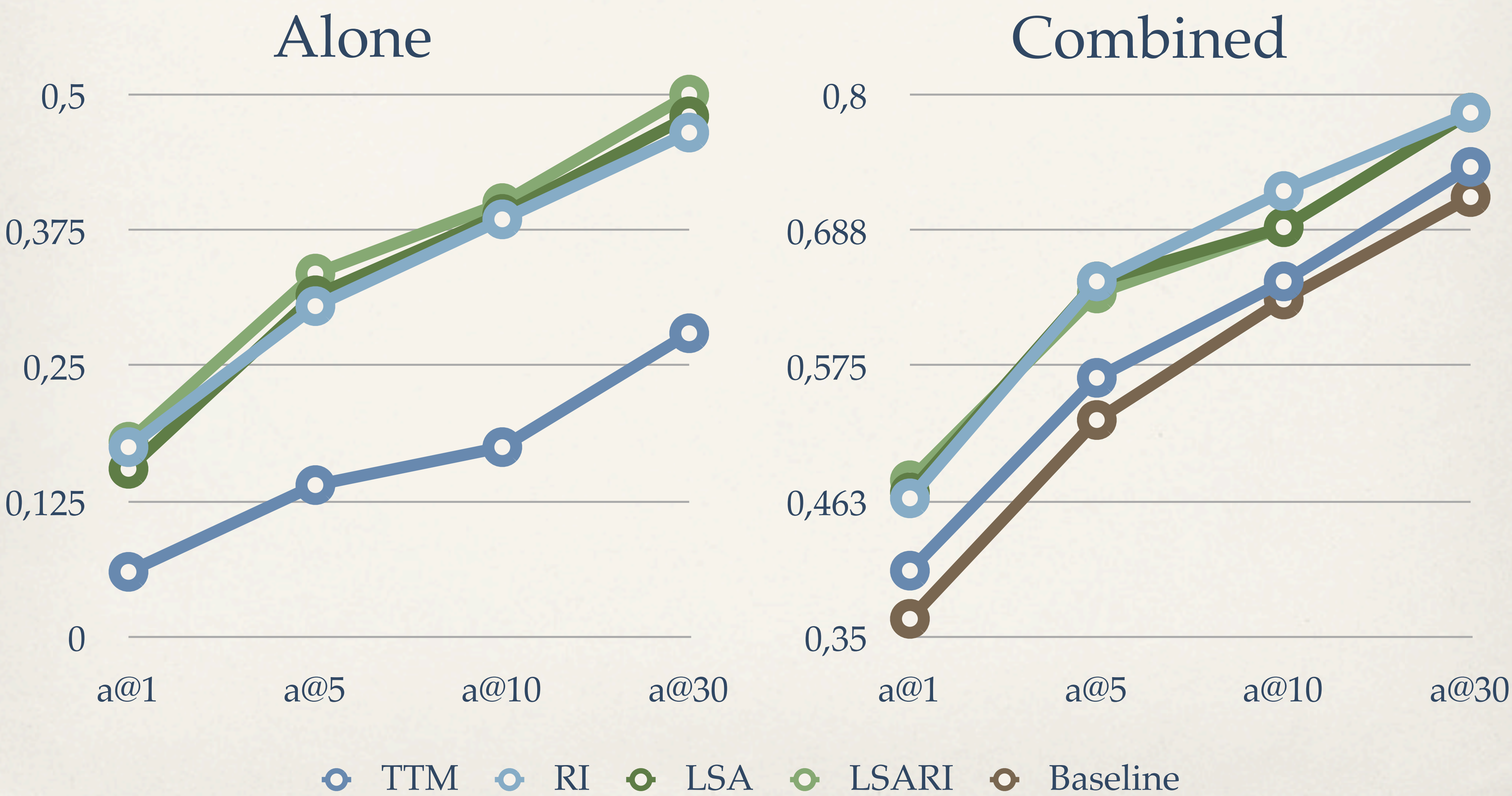
Results (English)

	Run	a@1	a@5	a@10	a@30	MRR
alone	TTM	0.060	0.145	0.215	0.345	0.107
	RI	0.180	0.370	0.425	0.535	0.267‡
	LSA	**0.205**	**0.415**	**0.490**	0.600	**0.300‡**
	LSARI	0.190	0.405	**0.490**	**0.620**	0.295‡
combined	baseline	0.445	0.635	0.690	0.780	0.549
	TTM	0.535	0.715	0.775	0.810	0.6141
	RI	0.550	**0.730**	0.785	**0.870**	**0.637†‡**
	LSA	**0.560**	0.725	**0.790**	0.855	**0.6371†**
	LSARI	0.555	**0.730**	**0.790**	**0.870**	0.6341†

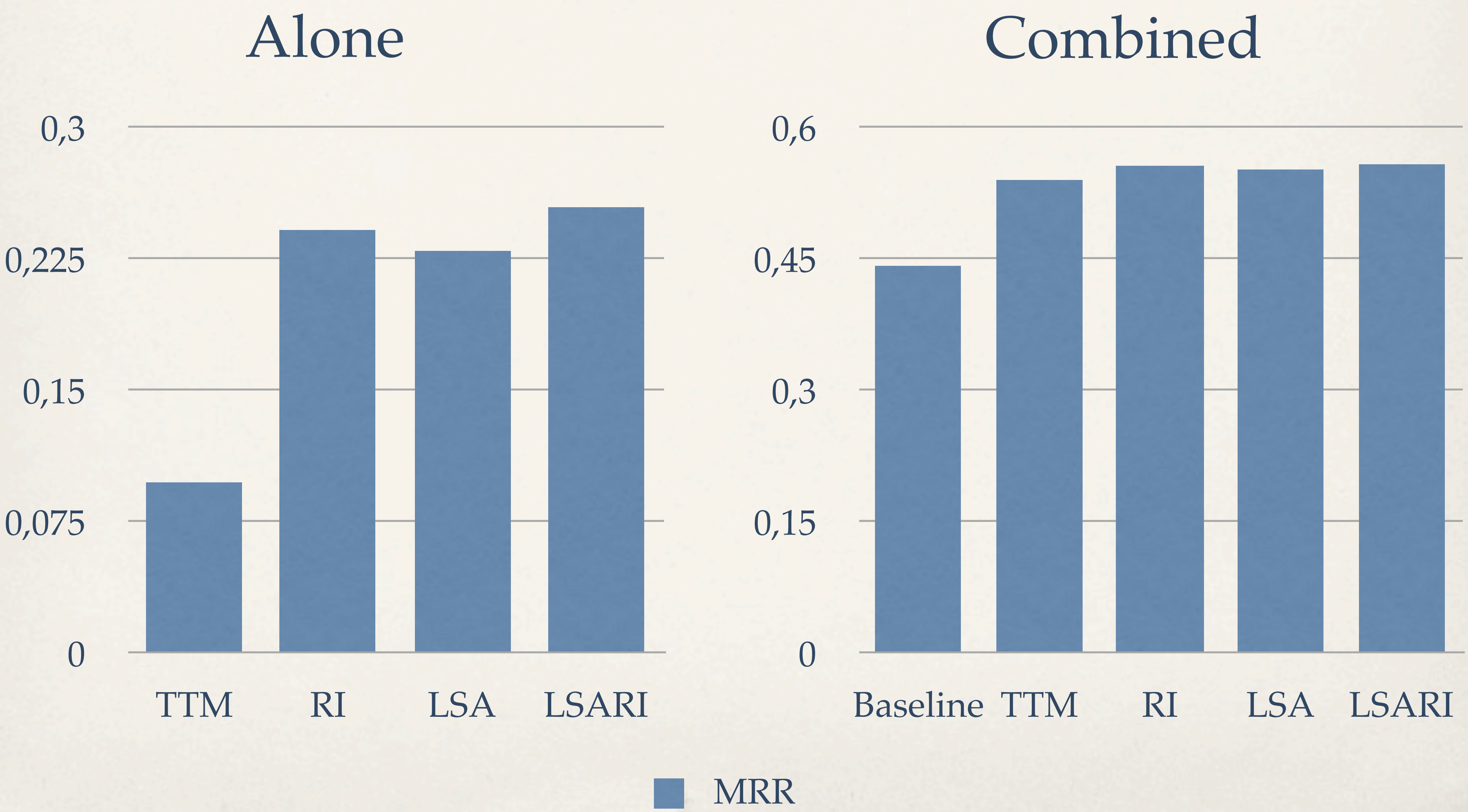
Significance wrt. the baseline (†)

Significance wrt. the TTM (‡)

Results (Italian) a@n



Results (Italian) MRR



Results (Italian)

	Run	a@1	a@5	a@10	a@30	MRR
alone	TTM	0.060	0.140	0.175	0.280	0.097
	RI	0.175	0.305	0.385	0.465	0.241‡
	LSA	0.155	0.315	0.390	0.480	0.229‡
	LSARI	**0.180**	**0.335**	**0.400**	**0.500**	**0.254**‡
combined	baseline	0.365	0.530	0.630	0.715	0.441
	TTM	0.405	0.565	0.645	0.740	0.5391†
	RI	0.465	**0.645**	**0.720**	**0.785**	0.5551†
	LSA	0.470	**0.645**	0.690	**0.785**	0.5511†
	LSARI	**0.480**	0.635	0.690	**0.785**	**0.557**†‡

Significance wrt. the baseline (†)

Significance wrt. the TTM (‡)

What we found out

- Alone: all the proposed DSMs **perform better** than the TTM, in particular LSA and LSARI
- Combined: all the combinations **overcome the baseline**
- English **+16%** (RI/LSA) - Italian **+26%** (LSARI)
- **No** remarkable **difference** in performance between LSA and LSARI
- Gives some evidence that **DSMs** can be **useful** for **answer re-ranking**

Learning to Rank experiment

- Similarity scorers' output as **features**
- **RankNet** - 100 epochs, 1 hidden layer, 10 hidden nodes, 0.005 learning rate
- 10 fold Cross Validation
- MRR **0.68** for English and **0.605** for Italian obtained with the LSARI DSM, **~10%** improvement

Future Work

- **Add more** IR-based, linguistic and Machine Translation based **features**
- More **composition operators** for DSMs
- Add other **semantic features** (LDA, NNMF, ESA, ...)
- More **extensive experiment** with parameter tuning, different MLR algorithms and different dataset

Thank you for your attention