Semantic Models for Question Answering

Piero Molino

Question Answering

Query = Natural Language Question

Result = Exact Answer or Short Passage

Q: Who’s the adoptive son of Julius Cesar?

A: 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**

Outline

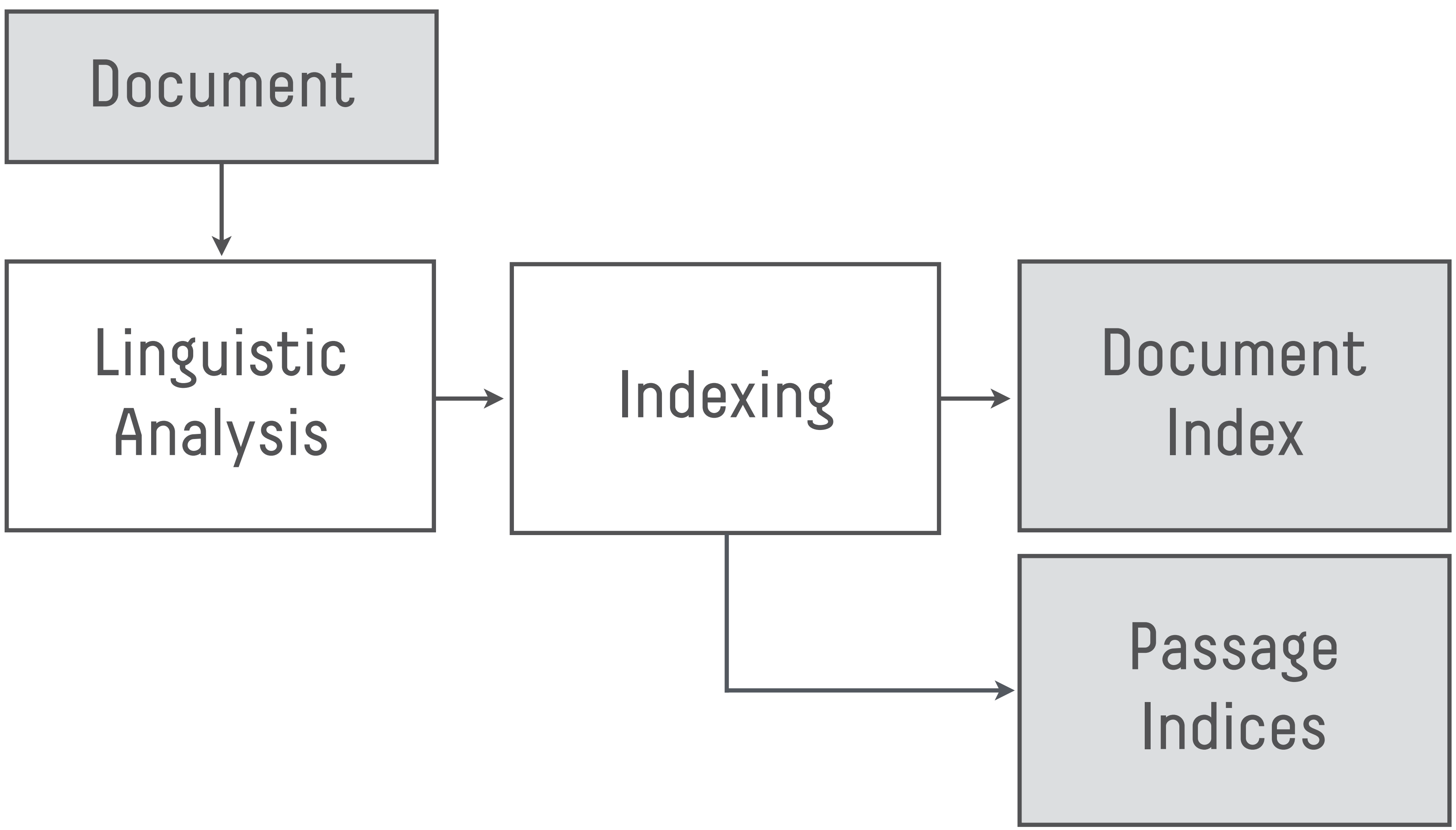
Introduction and Motivation

Distributional Semantics

Yahoo! Answers Experiment

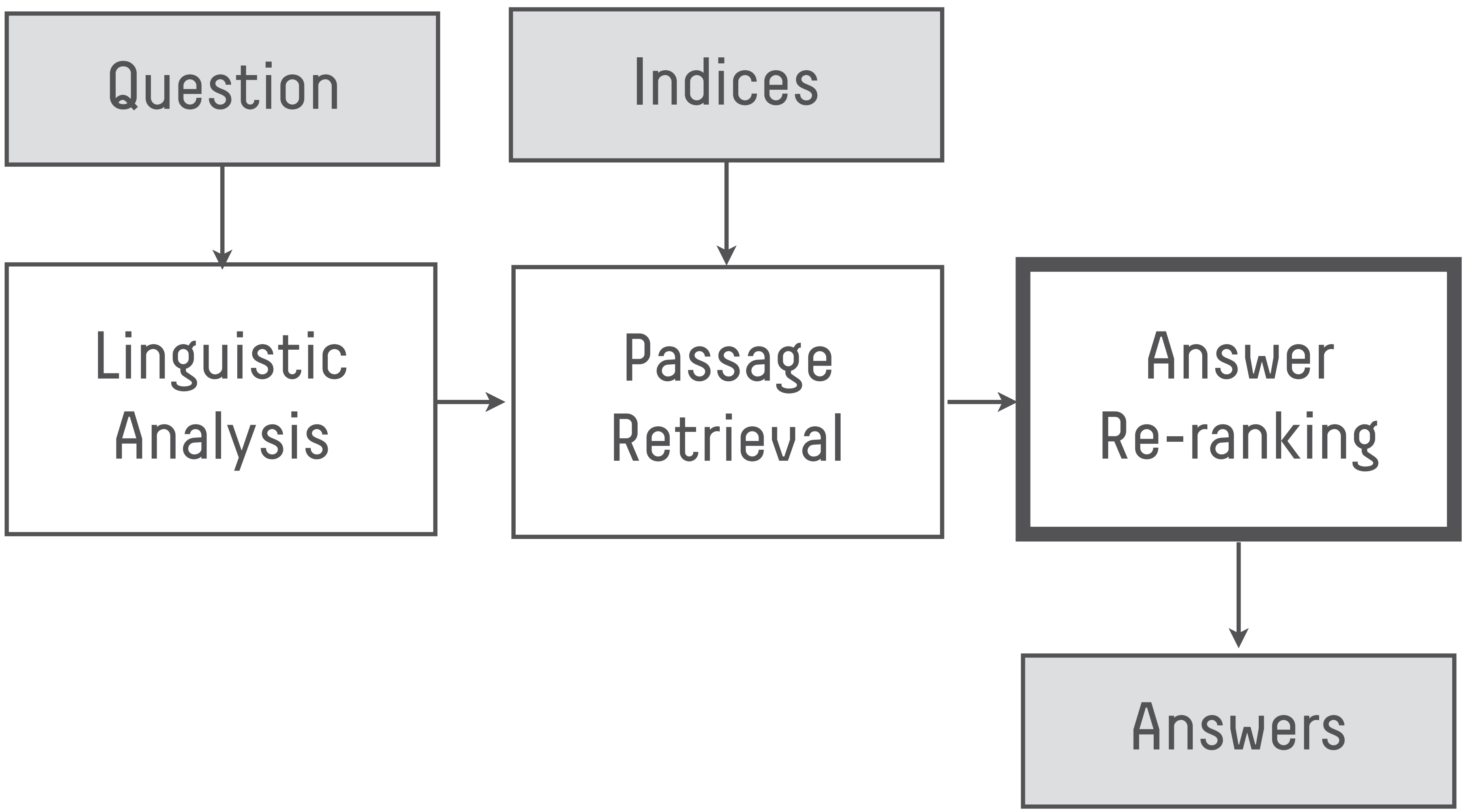
"Who Wants to Be Millionaire?" Experiment

General Architecture
Indexing



General Architecture

Retrieval



Learning to Rank

Learn the Ranking Function from Question-Answer

Represent Question-Answer pair as a datapoint with

- **Question specific** and **Answer specific** features (length, category, type of origin document, ...)

- **Question-Answer features** (different similarity measures, TFIDF, BM25, N-gram overlap, Machine Translation, syntactic similarity, ...)

Why semantics?

Some questions do not share even a **single word** with the answer

Q: Which beverages contain alcohol?

A: Wine makes you drunk

Ranking answers according to their **semantic similarity** with the question can overcome the problem

Distributional Semantic Models

Exploit **latent** or **explicit concepts** rather than words

Tasks:

semantic text similarity

synonyms detection

query expansion

topic identification

...

Models:

Latent Semantic Analysis

Random Indexing

Continuous Skip-gram Model

Non-negative Matrix Factorization

Latent Dirichlet Allocation

Explicit Semantic Analysis

Research Questions

RQ1 Are the distributional semantic representations good representations for the meaning of questions and answers?

RQ2 Can distributional semantic representations be combined with other criteria in order to obtain a better ranking of the answers?

Distributional Semantic Models

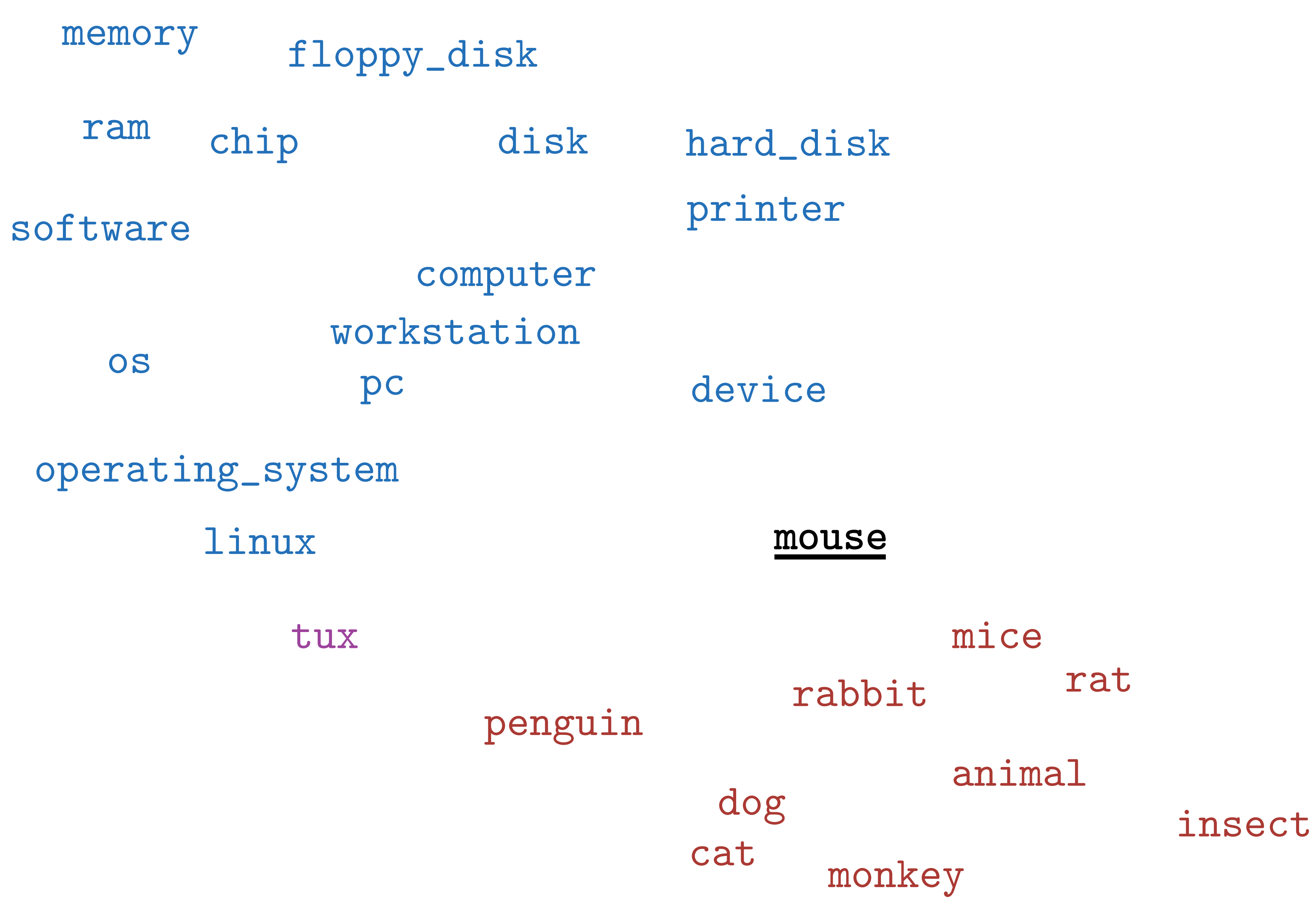
Represent words as points in a geometric space

Do not require specific text operations (corpus / language independent)

Widely used in IR and Computational Linguistic

Never been used for answer re-ranking

Distributional Semantic Models



Insight

Semantic similarity between **Question** and **Answer**

Computed with **Distributional Semantic Models**

Used as **re-rank feature**

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

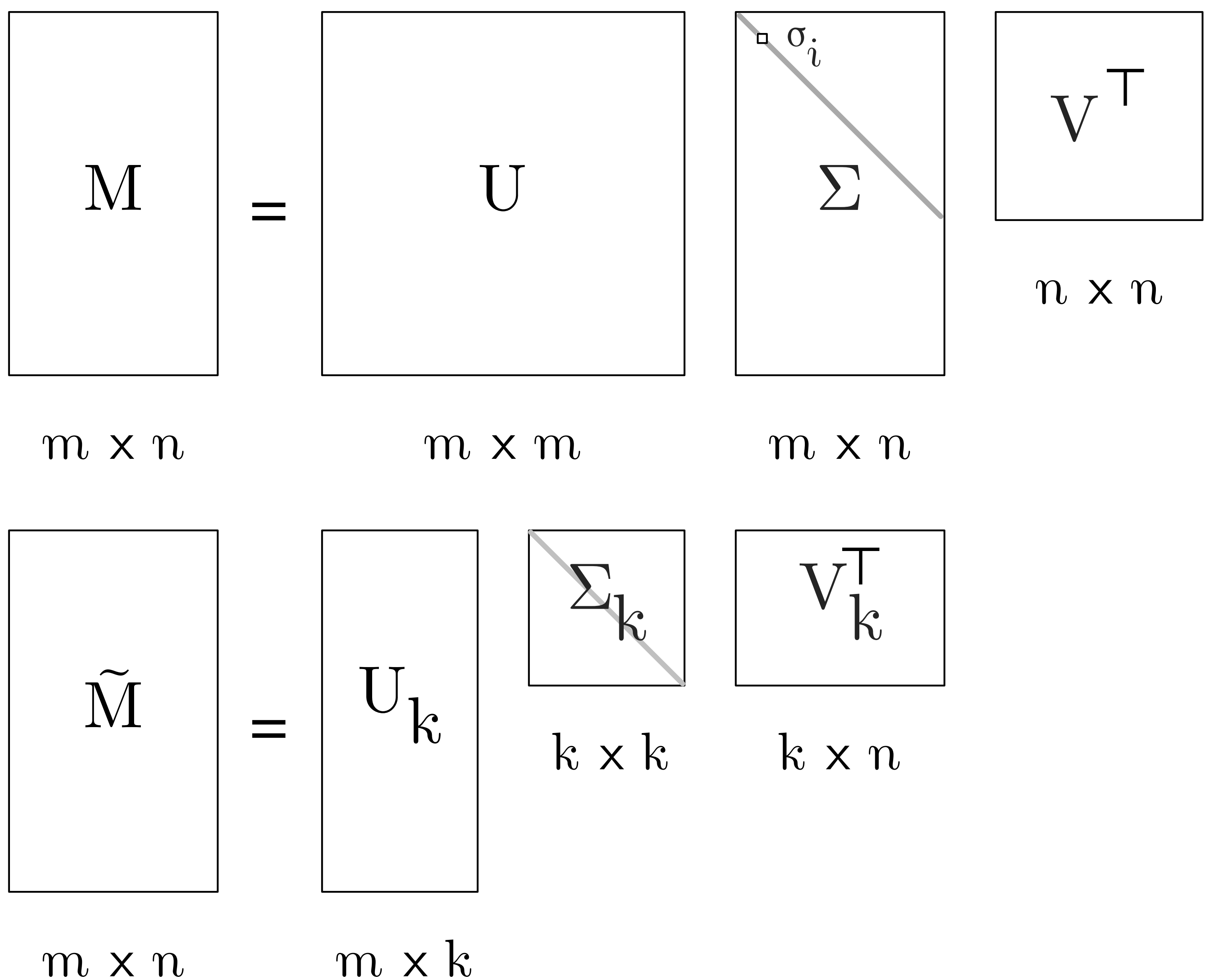
Latent Semantic Analysis (LSA): TSVD of the co-occurrence matrix

Random Indexing (RI): based on the Random Projection

Latent Semantic Analysis over Random Indexing (LSARI)

Continuous Skip-gram Model (CSGM): based on neural networks

Latent Semantic Analysis

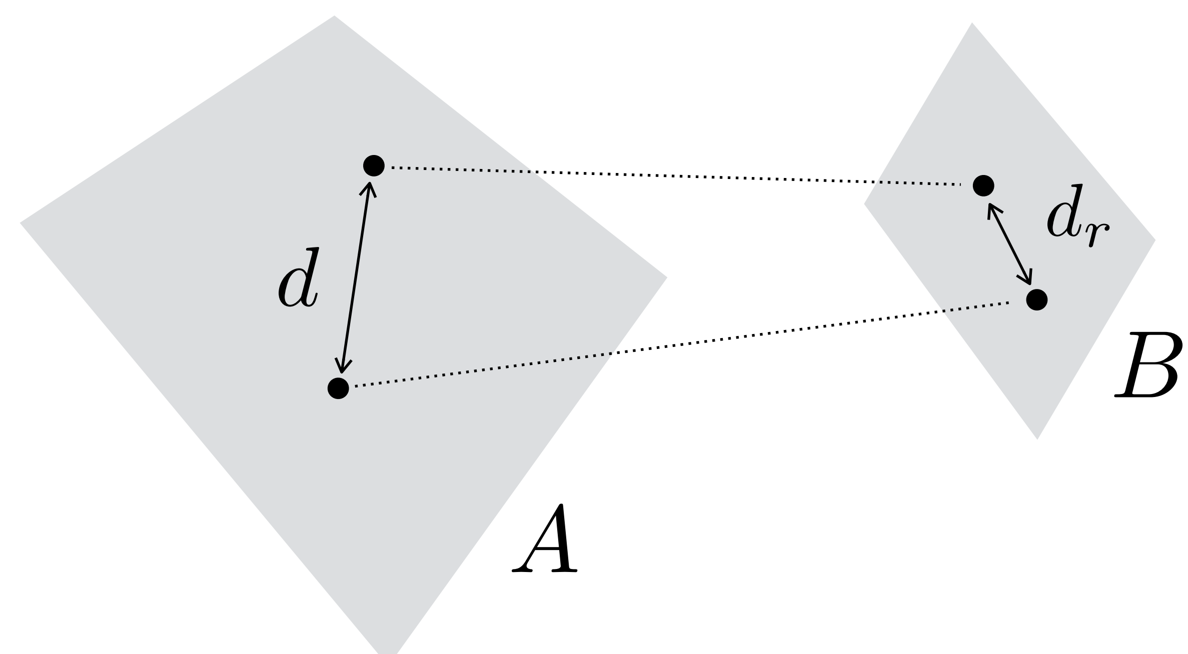


Random Indexing

Locality-sensitive hashing method which **approximate** the **distance** between points

$$B^{n,k} \approx A^{n,m} R^{m,k} \quad k \ll m$$

B preserves the euclidean distance between points in **A** (Johnson-Lindenstrauss lemma)



$$(1-\epsilon) d_r(v,u) \leq d(v,u) \leq (1+\epsilon) d_r(v,u)$$

Random Indexing

Construction

Generate and **assign** a Context Vector to each context element (e.g. document, passage, term, ...) with K random values in {-1, 0, +1} with **enforced sparsity**

Term Vector is the **sum** of the Context Vectors of all contexts in which the term **occurs**

Random Indexing

Example

Dataset: I drink **wine**

You drink **wine** and beer

Context Vectors

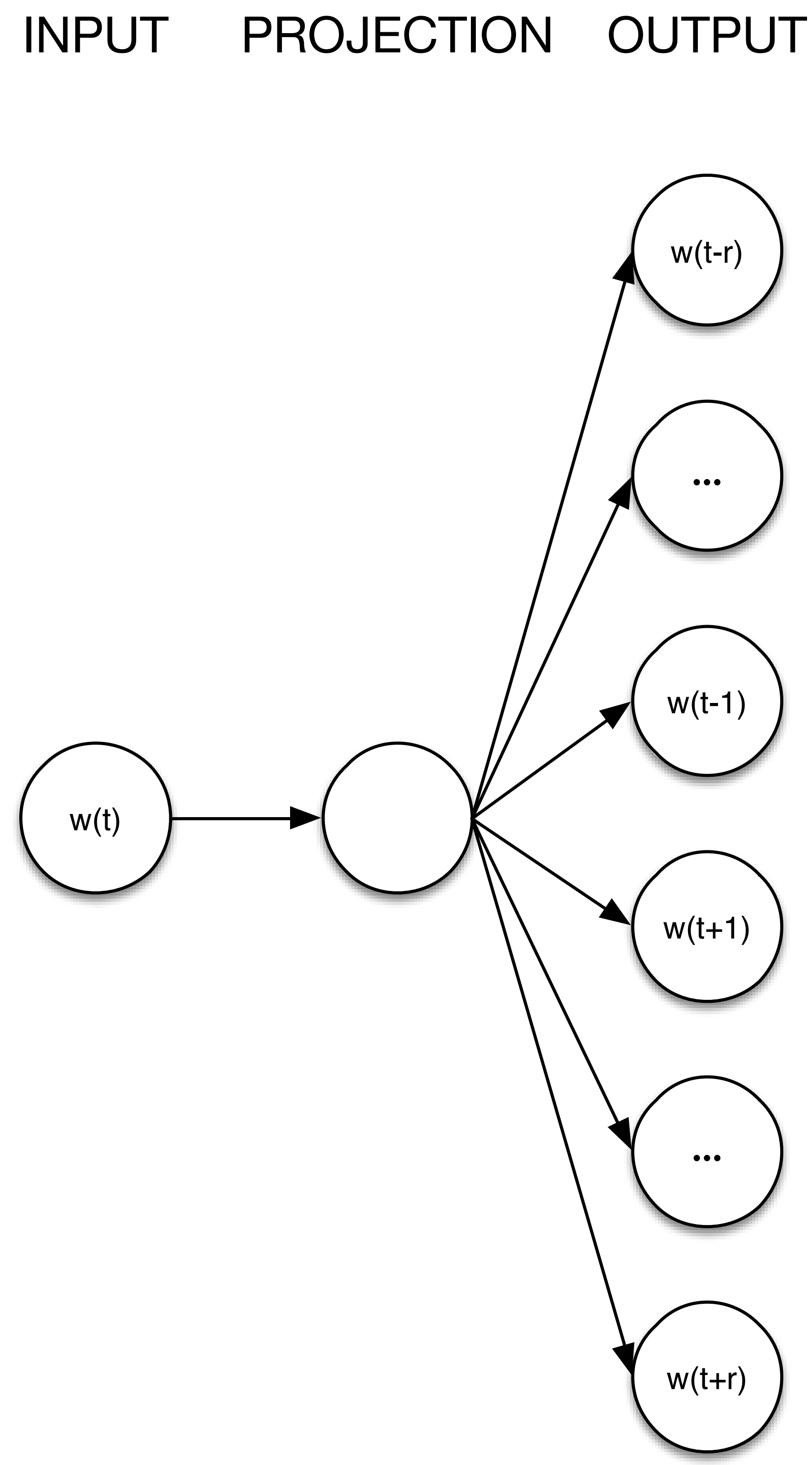
i	1	0	0	0	0	-1	0
drink	0	0	1	0	0	0	0
wine	0	1	0	0	0	0	0
you	0	-1	0	0	0	0	1
and	0	0	0	1	0	0	0
beer	-1	0	0	0	1	0	0

Term Vector for **wine**

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

wine	0	-1	2	0	1	-1	1

Continuous Skip-gram



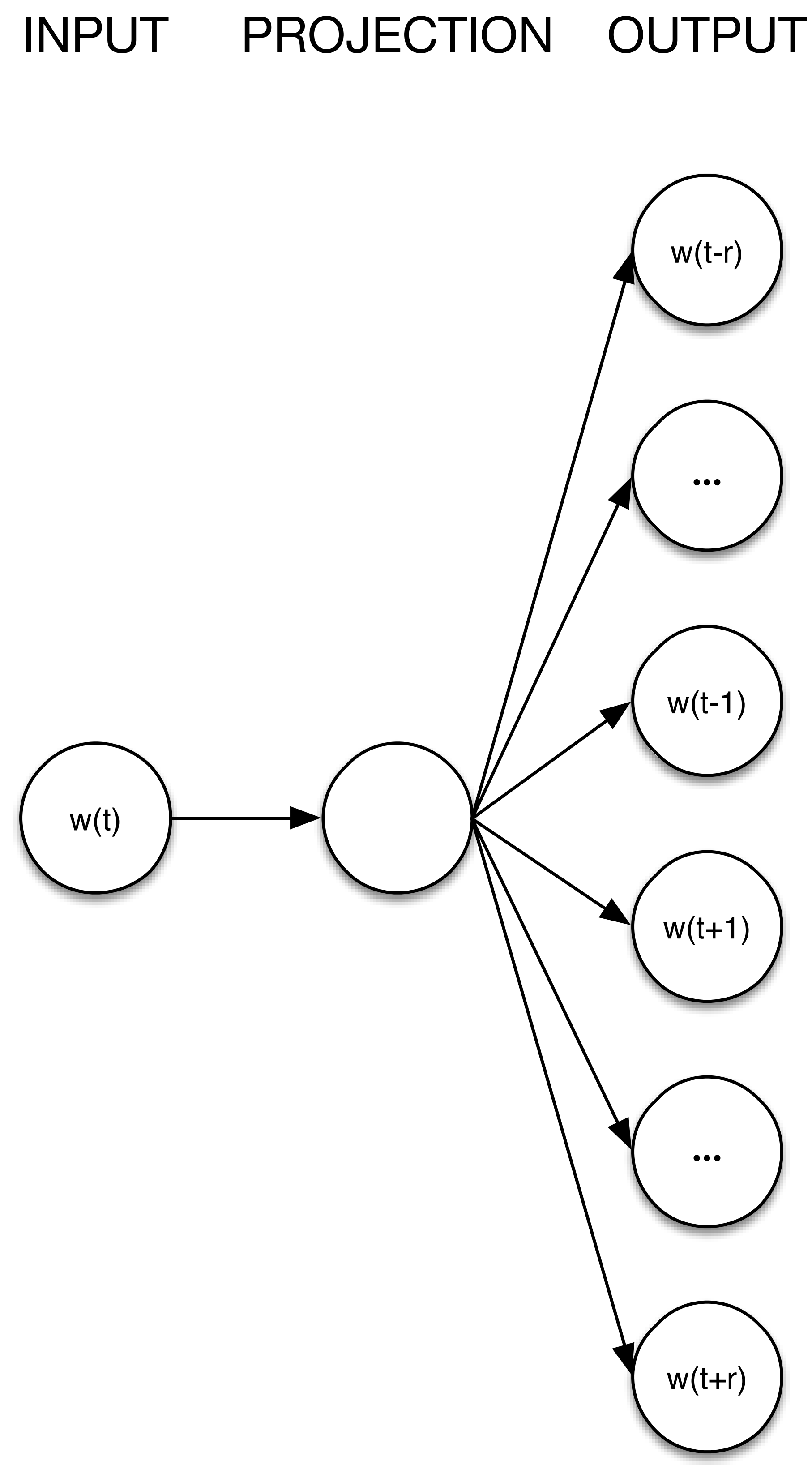
Feedforward Neural Network <u>without</u> hidden layer

Iterates over the words in the dataset, each word $w(t)$ is an input to a log-linear classifier with a continuous projection layer

Continuous Skip-gram

The output is a prediction of the words within a certain range before and after the input word

c is the fixed range before and after a word, a value *r* is obtained picking randomly a value between *[1,c]*



Compositionality

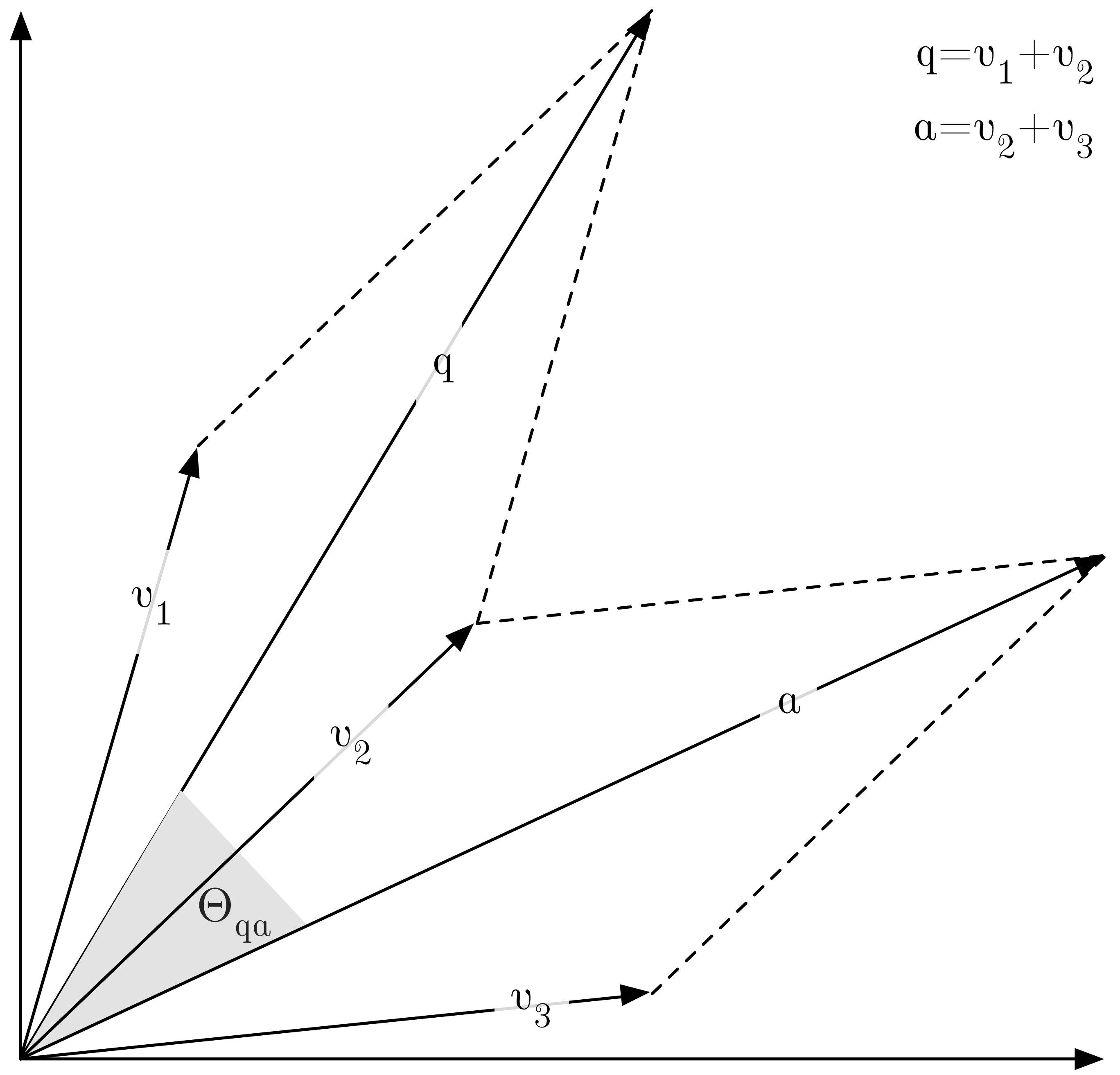
We need a method to represent question and answers, as they are **composed** by more than one term

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, ...) with no clear advantage

Compositionality



Yahoo! Answers Experiment

Best answer prediction on Yahoo! Answers data

2 Datasets

~220 features from different families:

- textual / content based
- user based
- network based

Lexicalizations

Different **lexicalization chains** (term, stem, lemma, lemma+pos, named entity, dependency, semantic role, supersense)

E.g. <u>John plays piano</u>

term, length 2: john-plays, plays-piano

lemma, length 1: john, play, piano

dependency (lemma), length 2: john-(subj)->play, piano-(dobj)-> play

semantic role (supersense), length 2: noun.person-A0->verb.perform, noun.artifact-A1->verb.perform

Textual features

Linguistic similarity (Overlap, Frequency, Density, Machine Translation, Length and Exact Sequence for all lexicalizations)

Text quality features (Visual Properties, Readability, Informativeness)

Distributional Semantics (LSA, RI, RILSA, CSGM on Wikipedia and answer corpus)

User features

User profile

Question and answers **counts and ratios**

Question and answers **counts and ratios per category**

Behavior (engagement)

Network Features

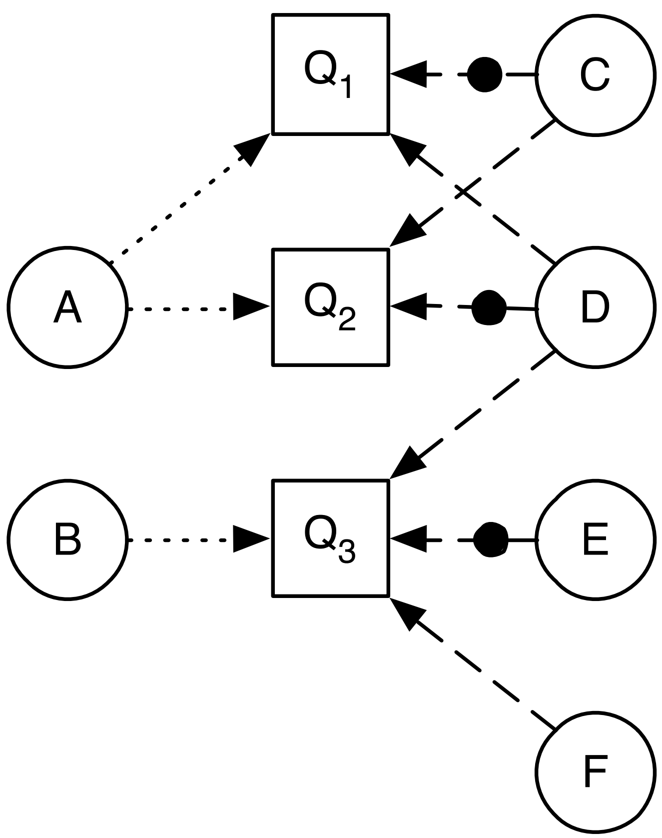
In-degree, **Hits authority** and **PageRank**

on 3 different networks:

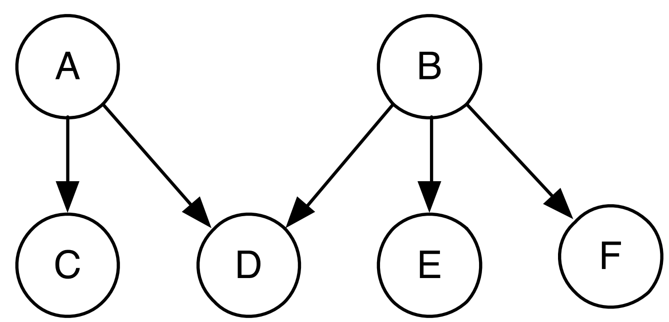
- **Asker-Replier**
- **Asker-Best-Answer**
- **Competition-Based-Expertise**

Network Features

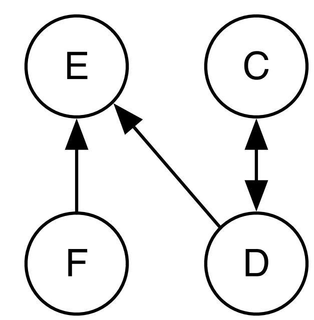
Question Answering Network



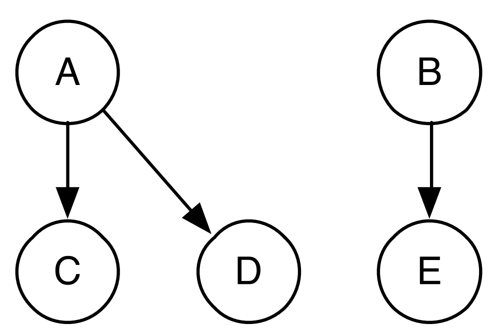
Asker-Replier Network

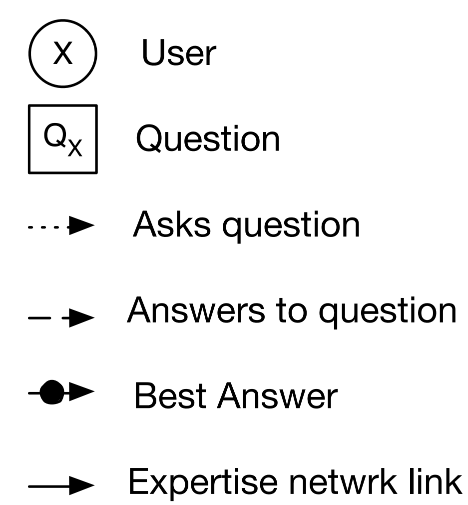


Competition-Based
Expertise Network



Asker Best Answer Network





Experimental Setting

Learning to Rank algorithm: **Random Forests**

Measures: **P@1, MRR, NDCG**

$$\mathrm{P@1} = rel_1$$

$$\mathrm{RR} = \frac{1}{\mathrm{rank(BA)}}$$

$$\mathrm{DCG_k} = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

rel_i is an indicator function returns *1* if the answer in the *i^{th}* position is the best answer

Dataset 1

Yahoo! Answers 2011 English questions

>7M questions >39M answers >6M users

Questions are clustered with k-means in **4 clusters**

factual-information seeking (31%)

subjective-information seeking (32%)

social discussion (10%)

poll-survey (27%)

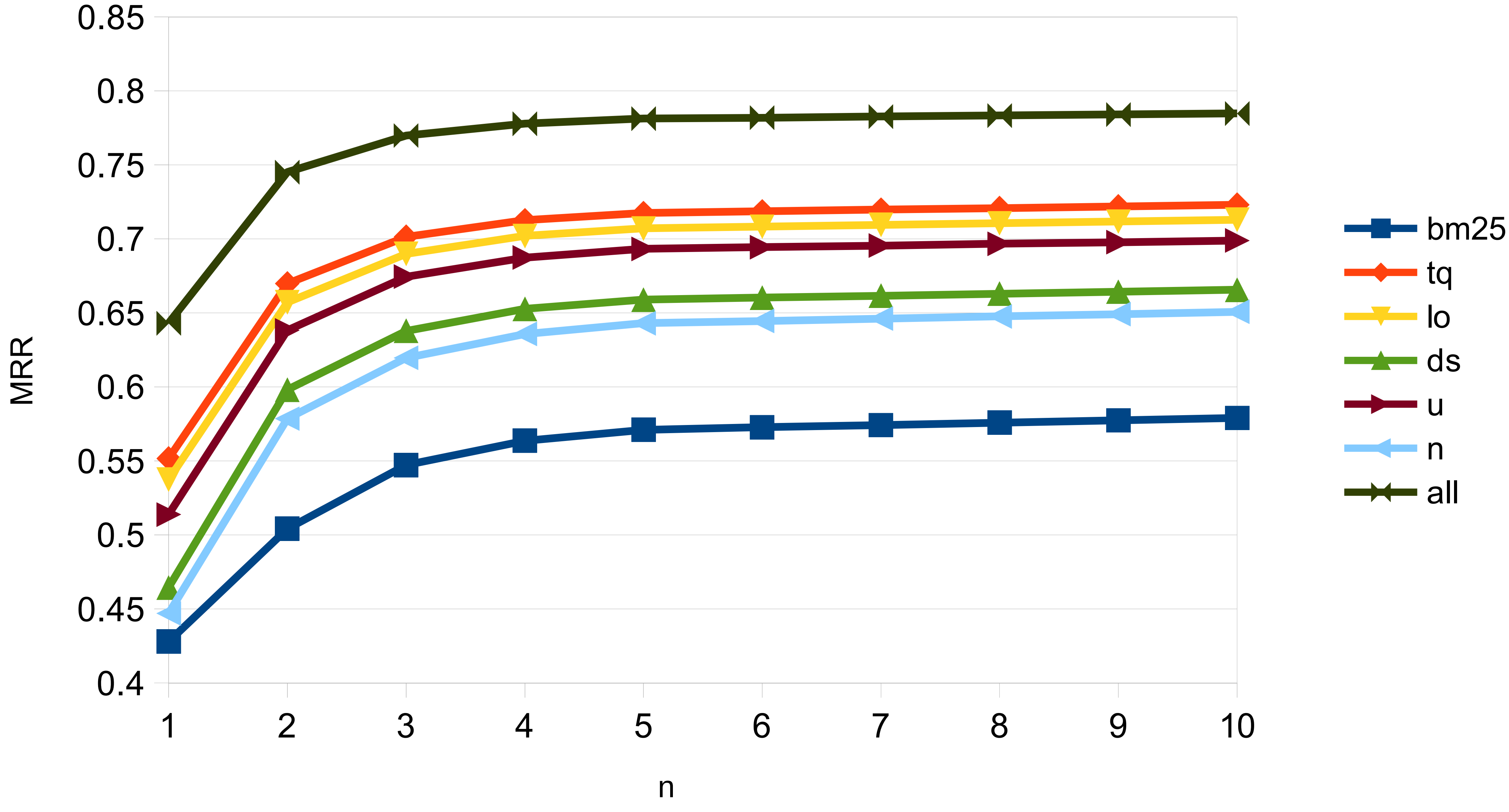
70-10-20 split based on timestamp

Results

Features	P@1	MRR	DCG
BM25	0.4143	0.5532	0.6567
Agichtein et al. [2008]	0.5243	0.6375	0.6962
tq	0.5305	0.7016	0.7655
ls	0.5143	0.6921	0.7613
ds	0.4782	0.6760	0.7564
u	0.5218	0.7009	0.7757
n	0.4527	0.6645	0.7484
tq+u	0.6201	0.7597	0.8260
tq+n	0.5862	0.7366	0.8080
tq+ds	0.5536	0.7144	0.7910
tq+ls	0.5515	0.7129	0.7897
tq+u+n	0.6416	0.7742	0.8370
tq+u+ds	0.6210	0.7606	0.8266
tq+u+ls	0.6199	0.7597	0.8260
tq+lo+ds	0.5519	0.7143	0.7901
tq+u+n+ds	0.6450	0.7752	0.8379
tq+u+n+ls	0.6414	0.7739	0.8368
all	0.6471	0.7798	0.8389

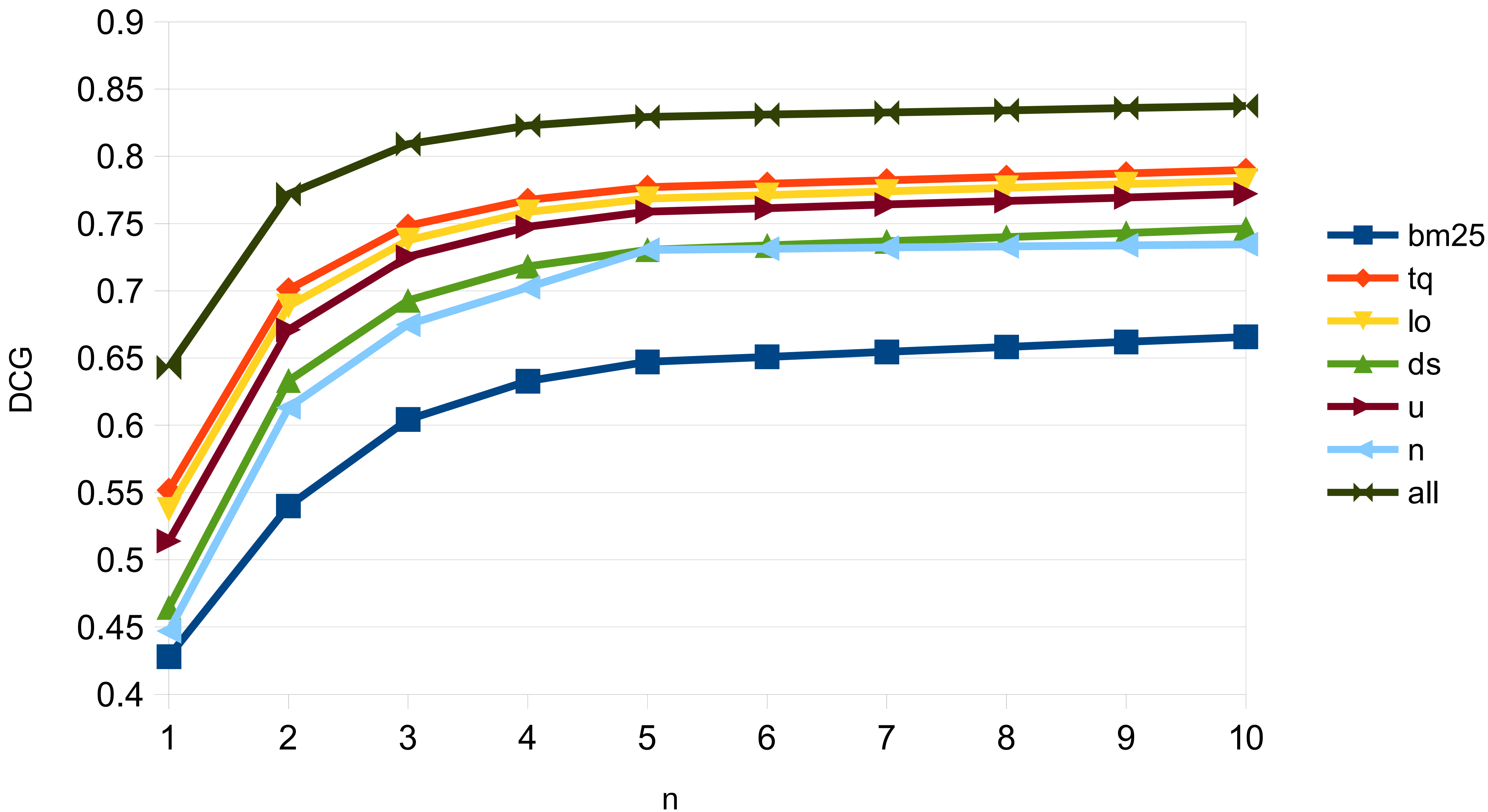
tq text quality **ls** linguistic similarity **ds** distributional semantics **u** user **n** network

MRR trends



MRR
n
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
1
2
3
4
5
6
7
8
9
10
bm25
tq
lo
ds
u
n
all

NDCG trends


0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
DCG
1
2
3
4
5
6
7
8
9
10
n
bm25
tq
lo
ds
u
n
all

Feature Ablation

Feature	**-Δ**		
tq: Preposition Count	0.049	**tq**: Conjunctions Count	0.035
tq: Verbs not in Question	0.045	**tq**: Capitalized Words Count	0.035
tq: Nouns not in Question	0.045	**tq**: "To be" Count	0.035
tq: Unique Words in Answer	0.043	**ls**: Lemma Overlap	0.034
tq: Pronouns Count	0.042	**ls**: Stem Overlap	0.034
tq: Punctuation Count	0.039	**ls**: Term Overlap	0.032
tq: Average Words per Sentence	0.039	**tq**: Auxiliary Verbs Count	0.034
ds: Random Indexing on Yahoo! Answers	0.039	**ls**: Super-senses BM25	0.031
ls: Super-senses Overlap	0.038	**n**: Indegree on CBEN	0.030
tq: Adjectives not in Question	0.036	**u**: Answerer's Best Answer Ratio	0.030

Distributional Features

Feature	**Rank**
ds: Random Indexing on Yahoo! Answers	8
ds: Continuous Skip-gram Model on Yahoo! Answers	30
ds: LSA on Wikipedia	37
ds: LSA after Random Indexing on Wikipedia	38
ds: Continuous Skip-gram Model on Wikipedia	39
ds: Random Indexing on Wikipedia	40
ds: LSA after Random Indexing on Yahoo! Answers	89
ds: LSA on Yahoo! Answers	90

Network Features

Feature	Rank
n: Indegree on CBEN	19
n: Hits on CBEN	32
n: Indegree on ABAN	101
n: Hits on ABAN	108
n: Indegree on ARN	161
n: Hits on ARN	164
n: PageRank on ARN	170
n: PageRank on CBEN	183
n: PageRank on ABAN	184

Clusters

	Factual	**Subjective**	**Discussion**	**Poll**
tq	**0.7329**	**0.7242**	0.6676	0.6762
ls	0.7243	0.7117	0.6482	0.6350
ds	0.6873	0.6732	0.6371	0.6492
u	0.7221	0.7118	**0.6724**	**0.6878**
n	0.7003	0.6953	0.6132	0.6214
all	0.8053	0.7892	0.7502	0.7638

Dataset 2

Yahoo! Answers Manner questions

142K questions and 771K answers

Match the regular expression

how (to | do | did | does | can | would | could | should), and have at least four words

<u>No information about the users</u>

Results

Features	P@1	MRR	DCG
BM25	0.4112	0.5606	0.6121
Surdeanu et al. [2011]	0.5091	0.6465	-
Hieber and Riezler [2011]	0.4844	0.6676	-
ds	0.6118	0.7689	0.8198
ls	0.618	0.7717	0.8236
tq	0.6245	0.7857	0.8352
ds+ls	0.618	0.7721	0.8236
ds+tq	**0.6532**	0.7920	0.8421
ls+tq	0.6401	0.7855	0.8352
ds+ls+tq	**0.6532**	**0.7922**	**0.8425**

tq text quality **ls** linguistic similarity **ds** distributional semantics

Different Ranking Algorithms

	LR	**RankSVM**	**ListNet**	**RF**
Manner	0.6952	0.7683	0.7520	**0.7922**
Factual	0.7407	0.7774	0.7626	**0.8059**
Subjective	0.7183	0.7640	0.7411	**0.7898**
Discussion	0.6881	0.7256	0.7059	**0.7508**
Poll	0.7027	0.7286	0.7312	**0.7644**
All	0.7165	0.7491	0.7466	**0.7798**

LR Logistic Regression **RF** Random Forests

Research Questions

RQ3 To what extent can a QA system be designed in a language-independent way, by preserving its effectiveness?

RQ4. Is it possible to develop an artificial player for the "Who Wants to Be a Millionaire?" game able to outperform human players?

Who Wants to Be Millionaire?

50:50

Who directed Blade Runner?

A **Harrison Ford**

B **Ridley Scott**

C **Philip Dick**

D **James Cameron**

Who Wants to Be Millionaire?

4 possible answers to each question

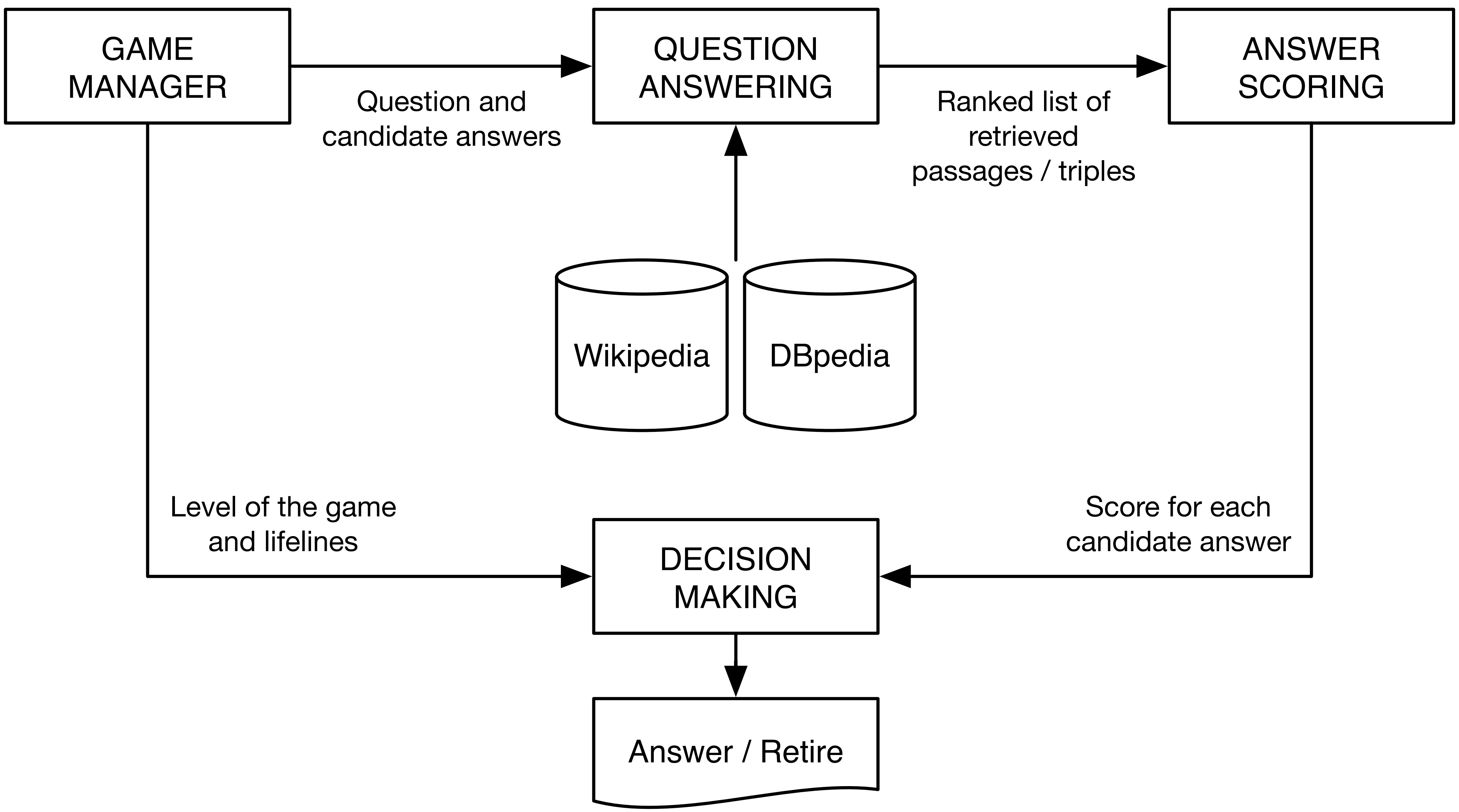
Choose a possible answer according to the results of a QA system

Answers are paragraphs obtained from Wikipedia or triples from DBpedia

Answers example

Article Title	Passage Text	Score
Ridley Scott	Sir Ridley Scott (born 30 November 1937) is an English film director and producer. Following his commercial breakthrough with Alien (1979), his best-known works are the sci-fi classic Blade Runner (1982) and the best picture Oscar-winner Gladiator (2000).	0.532
Blade Runner	Blade Runner is a 1982 American dystopian science fiction action film directed by Ridley Scott and starring Harrison Ford, Rutger Hauer, and Sean Young. The screenplay, written by Hampton Fancher and David Peoples, is loosely based on the novel Do Androids Dream of Electric Sheep? by Philip K. Dick.	0.510
Blade Runner	Director Ridley Scott and the film's producers "spent months" meeting and discussing the role with Dustin Hoffman, who eventually departed over differences in vision. Harrison Ford was ultimately chosen for several reasons.	0.500
Blade Runner	The screenplay by Hampton Fancher was optioned in 1977. Producer Michael Deeley became interested in Fancher's draft and convinced director Ridley Scott to film it.	0.490
Blade Runner	Interest in adapting Philip K. Dick's novel Do Androids Dream of Electric Sheep? developed shortly after its 1968 publication. Director Martin Scorsese was interested in filming the novel, but never optioned it.	0.120

Artificial Player Architecture



Decision Making

Conservative **heuristic rules** to manage the situations where:

- the **maximum confidence** for the four answers is low
- there is **no confidence** at all in the answers (when the passages are not helpful)
- the **difference** between the **maximum confidence** and the **second best confidence** is not large enough

Decide if to use a “lifeline”, to answer directly or to retire

Answer selection

Criteria	**Parameters**
Levenshtein	number of passages
Longest Common Subsequence	level of lexicalization
Term Overlap	stopword removal
Exact Substring	score of the passages
Density	question expansion
Distributional similarity	

Example feature: *TermOverlap* (2, Lemma, Yes, Yes, Yes)

Combination of **1200 features** with Random Forests

Experimental Setting

1960 Italian and 1960 English questions from the official WWBM board games, 5-fold cross validation

98 humans 20 questions each (only Italian)

Test the **accuracy** of the Answer Scoring

Google Baselines

1. Query **Google** and take **top 30 snippets**

2. Multiply the **number** of times the **answer occurred** in each snippet with the **inverse of the rank** of the snippet

Google Wikipedia Baseline: limit the results to Wikipedia articles for <u>fair comparison</u>

Best Single Criteria Italian

Rank	Criterion	P	Lex	S	SW	QE	Accuracy
1	Overlap	25	ST	Y	Y	N	64.29%
2	Overlap	25	LEM	Y	Y	N	64.29%
3	Density	3	KWD	Y	N	Y	64.03%
4	Density	30	ST	Y	Y	N	64.03%
5	Density	30	LEM	Y	Y	N	64.03%
6	Overlap	20	ST	Y	Y	N	63.78%
7	Overlap	20	LEM	Y	Y	N	63.78%
8	Overlap	30	ST	Y	Y	N	63.78%
9	Overlap	30	LEM	Y	Y	N	63.78%
10	Density	20	ST	Y	Y	N	63.27%
11	Density	20	LEM	Y	Y	N	63.27%
12	Density	25	KWD	Y	Y	N	63.01%
13	Overlap	15	ST	Y	Y	N	62.76%
14	Overlap	15	LEM	Y	Y	N	62.76%
15	Overlap	20	ST	N	Y	N	62.76%

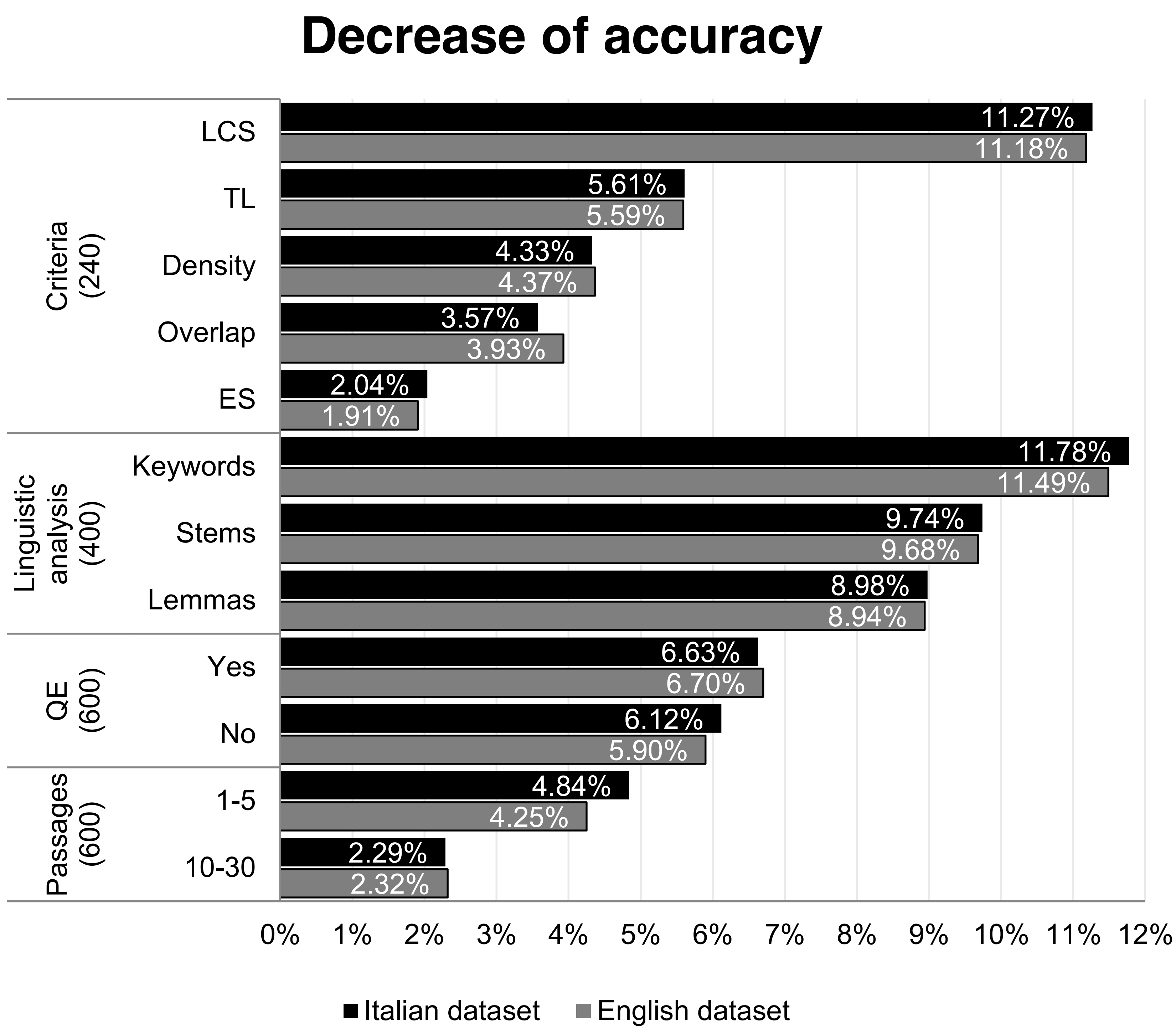
ST stem **LEM** lemma **KWD** keyword **S** score **SW** stopword **QE** question expansion

Best Single Criteria English

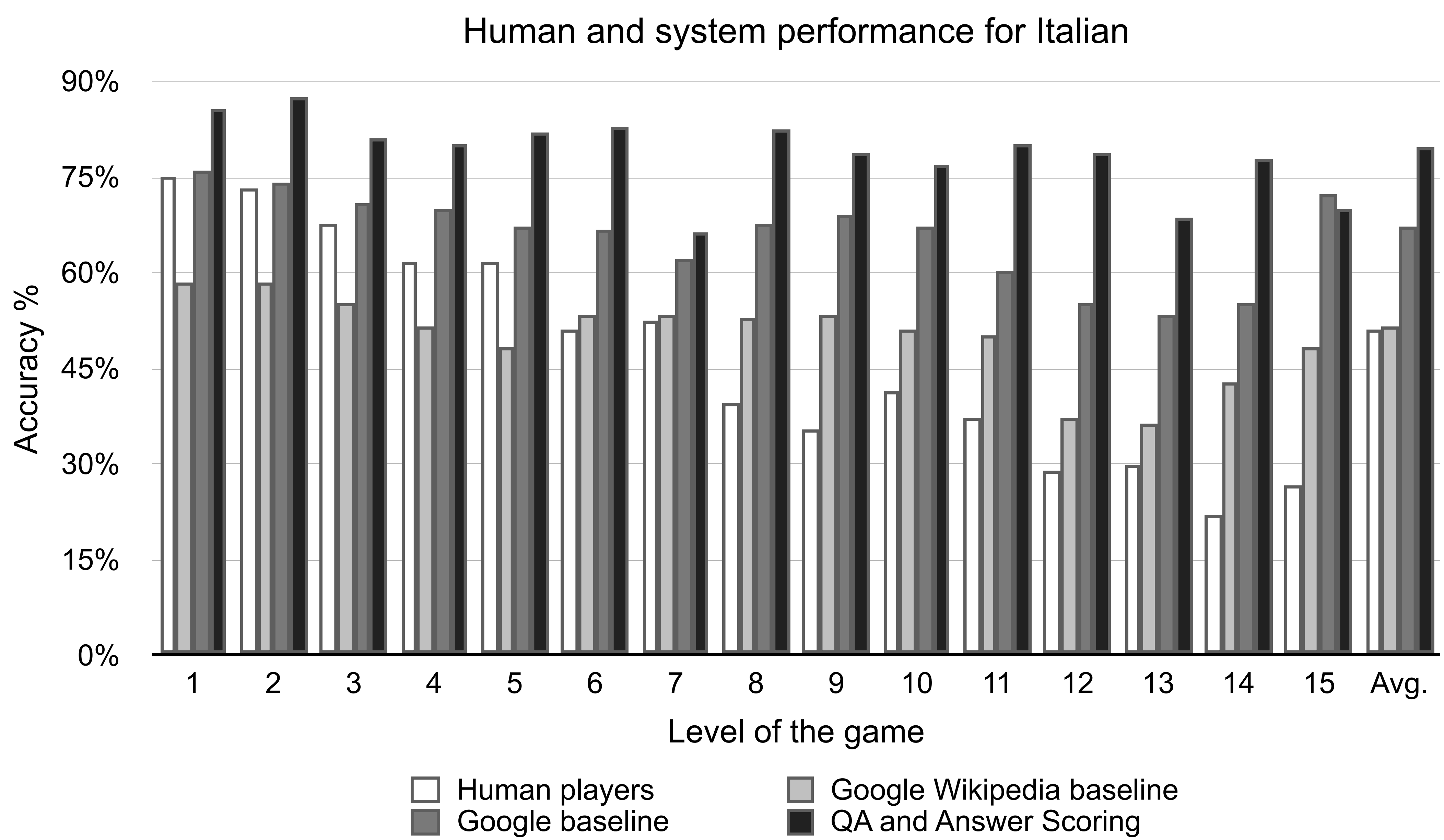
Rank	Criterion	P	Lex	S	SW	QE	Accuracy
1	Overlap	25	LEM	Y	Y	N	59.47%
2	Overlap	25	ST	Y	Y	N	59.38%
3	Density	3	KWD	Y	N	Y	59.26%
4	Overlap	20	ST	Y	Y	N	59.22%
5	Density	30	ST	Y	Y	N	59.08%
6	Density	30	LEM	Y	Y	N	59.08%
7	Overlap	30	ST	Y	Y	N	58.99%
8	Density	20	ST	Y	Y	N	58.84%
9	Overlap	15	LEM	Y	Y	N	58.72%
10	Density	25	KWD	Y	Y	N	58.37%
11	Overlap	30	LEM	Y	Y	N	58.35%
12	Density	20	LEM	Y	Y	N	58.21%
13	Overlap	20	LEM	Y	Y	N	58.14%
14	Overlap	20	ST	N	Y	N	57.99%
15	Overlap	15	ST	Y	Y	N	57.97%

ST stem **LEM** lemma **KWD** keyword **S** score **SW** stopword **QE** question expansion

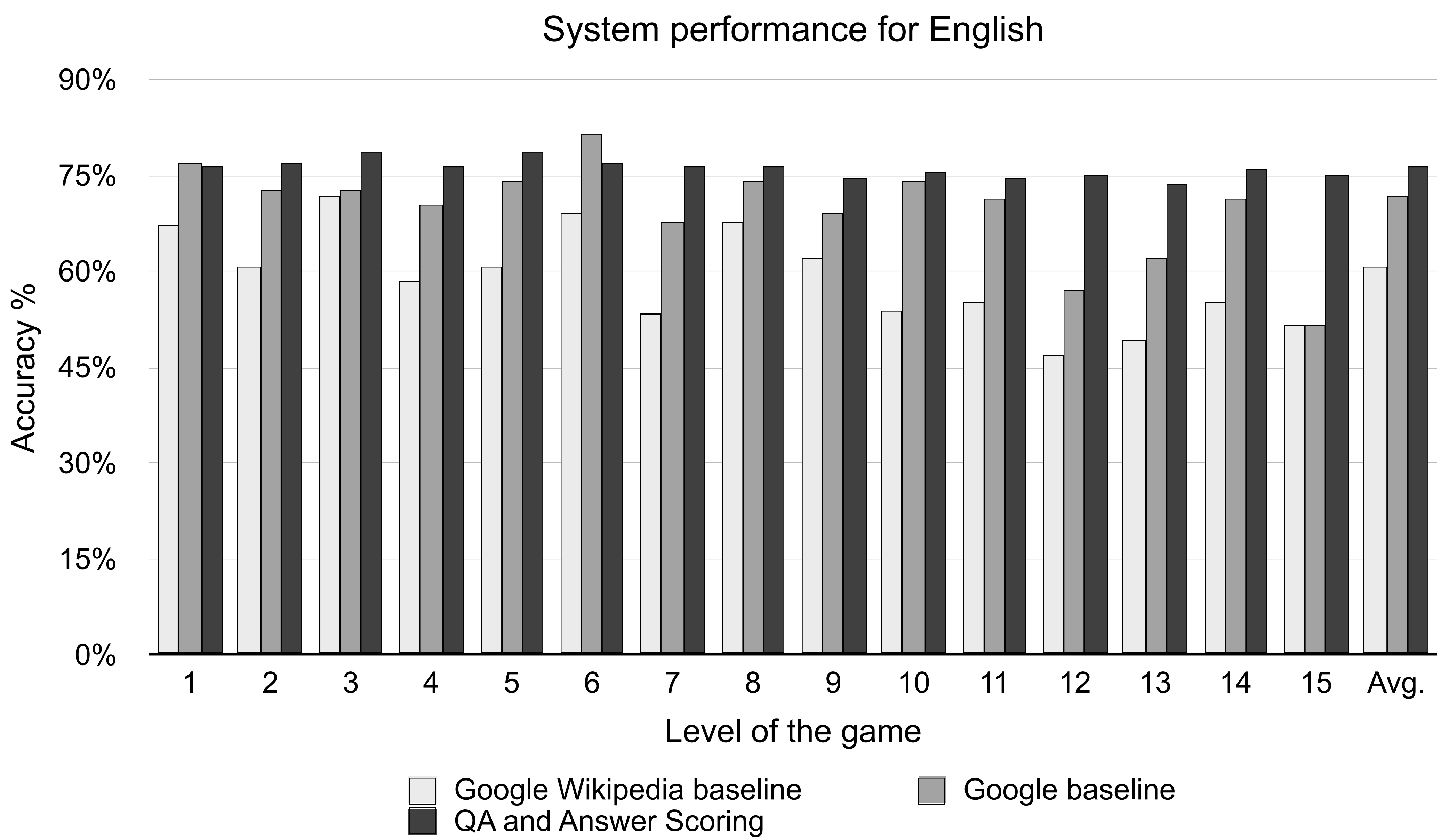
Feature Groups Ablation



Accuracy Italian



Accuracy English



Gameplay Experiment

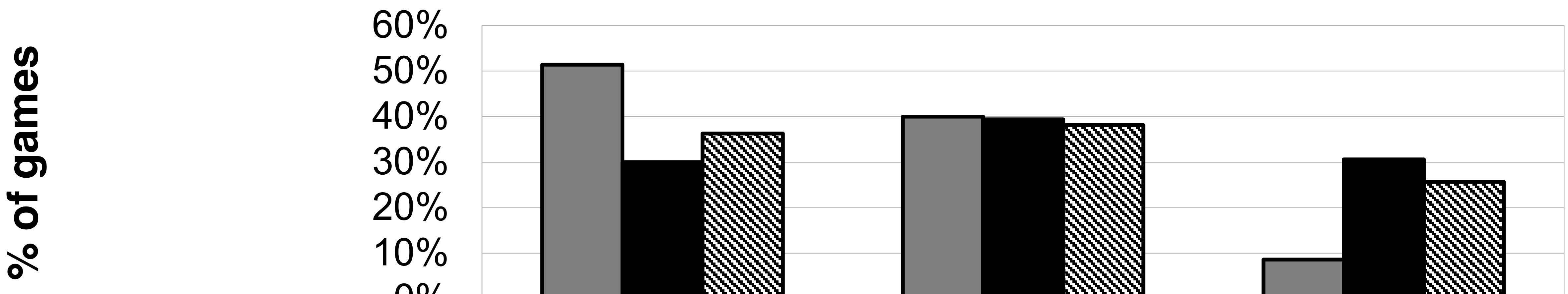
35 human players, 325 matches without overlapping questions for the same player

Test the ability of the **Artificial Player** (including **Decision Making**) in playing the game following its rules

Evaluated in terms of money earned and reached level

Reached level

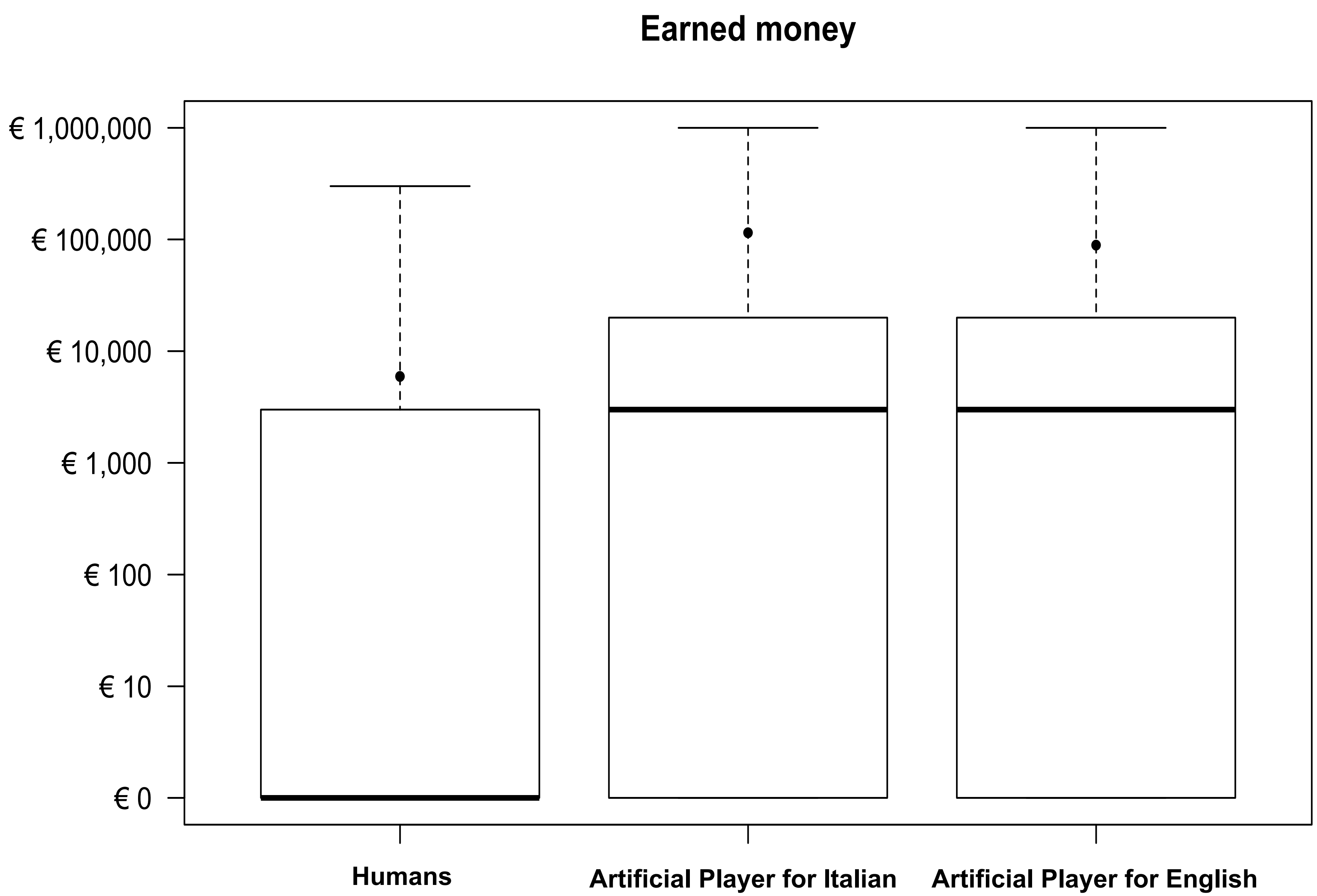
Level reached



	Level 1 to 5	Level 6 to 10	Level 11 to 15
Human Players	51.38%	40.00%	8.62%
Artificial Player for Italian	30.00%	39.38%	30.62%
Artificial Player for English	36.25%	38.13%	25.62%

The **Artificial Player wins** the game **17 times** for Italian and **12 times** for English, while **human players never win**

Earned Money



The **Artificial Player** earns on average **€114,531** for Italian and **€88,878** for English, while **human players** earn **€5,926**

Conclusions

RQ1 The new distributional semantics based features proposed achieve **surprisingly good results** considering their small number

RQ2 Distributional semantics based features **help achieving better ranking**. They are to be preferred to linguistic similarity ones as their contribution **overlaps** and they are **less computationally expensive**

Conclusions

RQ3 Definition of an **effective language-independent framework** for **QA** and **answer validation** leveraging open knowledge sources

RQ4 Built an **Artificial Player** which **outperforms human players** in terms of **average accuracy** and **money earned** playing WWBM

Thank you for your attention

THE BEST THESIS DEFENSE IS A GOOD THESIS OFFENSE.

Published papers

Piero Molino, Pasquale Lops, Giovanni Semeraro, Marco de Gemmis, Pierpaolo Basile. Playing with knowledge: A virtual player for "Who Wants to Be a Millionaire?" that leverages question answering techniques. Artificial Intelligence 222: 157-181 (2015)

Piero Molino, Luca Maria Aiello. Distributed Representations for Semantic Matching in non-factoid Question Answering. SMIR@SIGIR 2014: 38-45

Piero Molino, Gianvito Pio, Corrado Mencar. Fast Fuzzy Inference in Octave. Int. J. Computational Intelligence Systems 6(2): 307-317 (2013)

Piero Molino, Pierpaolo Basile, Ciro Santoro, Pasquale Lops, Marco de Gemmis, Giovanni Semeraro. A Virtual Player for "Who Wants to Be a Millionaire?" based on Question Answering. AI*IA 2013: 205-216

Piero Molino, Pierpaolo Basile, Annalina Caputo, Pasquale Lops, Giovanni Semeraro. Distributional Semantics for Answer Re-ranking in Question Answering. IIR 2013: 100-103

Piero Molino. Semantic models for answer re-ranking in question answering. SIGIR 2013: 1146-1147

Piero Molino, Pierpaolo Basile. QuestionCube: a Framework for Question Answering. IIR 2012: 167-178

Piero Molino, Pierpaolo Basile, Annalina Caputo, Pasquale Lops, Giovanni Semeraro. Exploiting Distributional Semantic Models in Question Answering. ICSC 2012: 146-153

Lifelines

50:50

Remove 2 wrong answers randomly

Poll the Audience

[50%,80%] correct *1st level*

[20%,35%] correct *15th level*

Phone a Friend

[1, 5] always correct

[6, 10] randomly correct or no answer

[11, 15] randomly correct, no answer or wrong answer

Decision Making Algorithm

Algorithm 2 Decision making algorithm

```
procedure DECISIONMAKING(< q, (c_A, c_B, c_C, c_D) >, lifelines)    ▷
    Decision strategy based on the scores of the four candidate answers for question q, and
    the available lifelines
    BestAnswer ← BEST(< q, (c_A, c_B, c_C, c_D) >)
    SecondBestAnswer ← SECONDBEST(< q, (c_A, c_B, c_C, c_D) >)
    if BestAnswer.score < threshold_1
    or (BestAnswer.score − SecondBestAnswer.score)
    < (BestAnswer.score ∗ threshold_2) then
        if CANUSE(Poll the Audience) then
            audienceAnswers ← USE(Poll the Audience)
            lifelines ← lifelines − {Poll the Audience}
            if audienceAnswers.score > threshold_1 then
                RETURN BEST(audienceAnswers)
            end if
        end if
        if CANUSE(Phone a Friend) then
            friendAnswer ← USE(Phone a Friend)
            lifelines ← lifelines − {Phone a Friend}
            if friendAnswer ≠ null then
                RETURN friendAnswer
            end if
        end if
        if (CANUSE(50:50) and CANRISK()) then
            50 : 50answers ← USE(50:50)
            lifelines ← lifelines − {50:50}
            if 50 : 50answers.score > threshold_1 then
                RETURN BEST(50:50answers)
            else
                RETURN RANDOM(50:50answers)
            end if
        end if
        if CANRISK() then
            RETURN RANDOM(answers)          ▷ No more lifelines but the player can risk
        end if
        RETIRE()
    else
        RETURN BestAnswer
    end if
end procedure
```

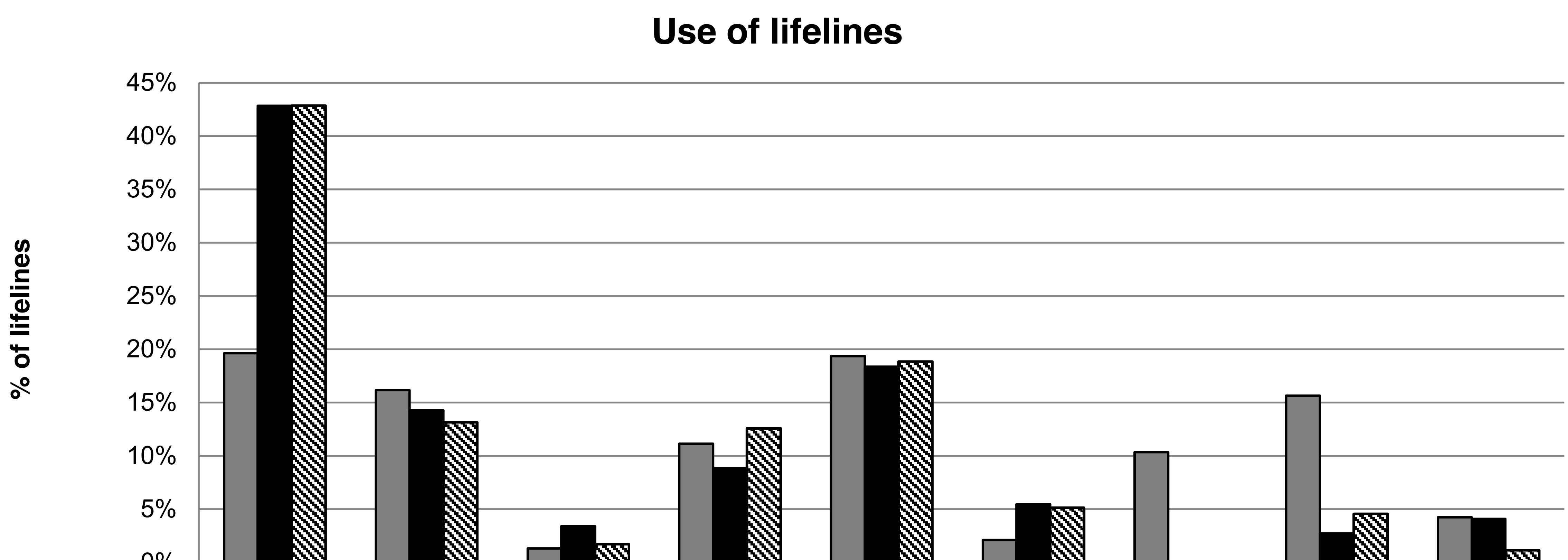
the **difference** between the **maximum confidence** and the **second best confidence** is not large enough

the **maximum confidence** for the four answers is low

there is **no confidence** at all in the answers

1. Poll the Audience
2. Phone a Friend
3. 50:50

Use of lifelines



	Poll the Audience			Phone a Friend			50-50		
	Level 1 to 5	Level 6 to 10	Level 11 to 15	Level 1 to 5	Level 6 to 10	Level 11 to 15	Level 1 to 5	Level 6 to 10	Level 11 to 15
Humans	19.63%	16.18%	1.33%	11.14%	19.36%	2.12%	10.34%	15.65%	4.24%
AP for Italian	42.86%	14.29%	3.40%	8.84%	18.37%	5.44%	0.00%	2.72%	4.08%
AP for English	42.86%	13.14%	1.71%	12.57%	18.86%	5.14%	0.00%	4.57%	1.14%

DBpedia

Indexing

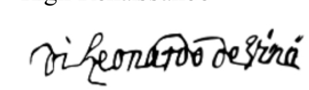
Manually created ~**350** questions tagged with **DBpedia properties** (top 50) to trained a **centroid classifier**

Create documents with the lexicalization of **RDF triples** with the <u>same subject</u>

Ex. ⟨Leonardo da Vinci, date of birth, 1452-04-15⟩

Leonardo da Vinci

Portrait of Leonardo by Melzi

Born	Leonardo di ser Piero da Vinci April 15, 1452 Vinci, Republic of Florence (present-day Italy)
Died	May 2, 1519 (aged 67) Amboise, Kingdom of France
Known for	Diverse fields of the arts and sciences
Notable work(s)	*Mona Lisa* *The Last Supper* *The Vitruvian Man* *Lady with an Ermine*
Style	High Renaissance
Signature	

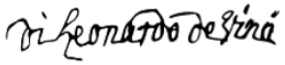
DBpedia
Retrieval

Question tagged with **DBpedia property** by the classifier and **Named Entities**

Extract from documents the list of passages (**RDF triples**) with the corresponding **DBpedia property** and **Named Entities**

Leonardo da Vinci

Portrait of Leonardo by Melzi

Born	Leonardo di ser Piero da Vinci April 15, 1452 Vinci, Republic of Florence (present-day Italy)
Died	May 2, 1519 (aged 67) Amboise, Kingdom of France
Known for	Diverse fields of the arts and sciences
Notable work(s)	*Mona Lisa* *The Last Supper* *The Vitruvian Man* *Lady with an Ermine*
Style	High Renaissance
Signature	

Preliminary Experiment

Dataset 2010 CLEF QA Competition

10.700 documents from European Union legislation and European Parliament transcriptions

200 questions in English and Italian

Metrics a@n (success@n) and MRR

Preliminary Experiment

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 for 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 ($\dagger$)

Significance wrt. the TTM ($\ddagger$)

Results for 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 (‡)

Thank you for your attention