Passage Retrieval & Re-ranking

Ling573
NLP Systems & Applications
April 18, 2013

Roadmap

- Passage retrieval vs passage ranking
  - Comparisons of
    - Passage unit size
    - Passage type
- Passage re-ranking
  - Exploiting deeper processing
    - Dependency matching
    - Answer types

Units of Retrieval

- *Simple is Best: Experiments with Different Document Segmentation Strategies for Passage Retrieval*
  - Tiedemann and Mur, 2008

Units of Retrieval

- *Simple is Best: Experiments with Different Document Segmentation Strategies for Passage Retrieval*
  - Tiedemann and Mur, 2008
  - Comparison of units for retrieval in QA
    - Documents
    - Paragraphs
    - Sentences
    - Semantically-based units (discourse segments)
    - Spans

Motivation

- Passage units necessary for QA
  - Focused sources for answers
  - Typically > 20 passage candidates yield poor QA

Motivation

- Passage units necessary for QA
  - Focused sources for answers
  - Typically > 20 passage candidates yield poor QA

- Retrieval fundamentally crucial

Motivation

- Passage units necessary for QA
  - Focused sources for answers
  - Typically > 20 passage candidates yield poor QA

- Retrieval fundamentally crucial
- Re-ranking passages is hard

Motivation

- Passage units necessary for QA
  - Focused sources for answers
  - Typically > 20 passage candidates yield poor QA
- Retrieval fundamentally crucial
- Re-ranking passages is hard
  - Tellex et al experiments
    - Improvements for passage reranking, but
    - Still dramatically lower than oracle retrieval rates

	Strict				
	Lucene		PRISE		TREC
Algorithm	MRR	% Inc.	MRR	% Inc.	% Inc.
IBM	0.326	49.20%	0.331	39.60%	44.3%
ISI	0.329	48.80%	0.287	41.80%	41.7%
SiteQ	0.323	48.00%	0.358	40.40%	56.1%
MultiText	0.354	46.40%	0.325	41.60%	43.1%
Alicante	0.296	50.00%	0.321	42.60%	60.4%
bm25	0.312	48.80%	0.252	46.00%	n/a
stemmed MITRE	0.250	52.60%	0.242	58.60%	n/a

Algorithm	# Incorrect	% Incorrect	MRR
IBM	31	7.18%	0.851
SiteQ	32	7.41%	0.859
ISI	37	8.56%	0.852
Alicante	39	9.03%	0.816
MultiText	44	10.19%	0.845
bm25	45	10.42%	0.810
MITRE	45	10.42%	0.800
stemmed MITRE	63	14.58%	0.762

Passages

- Some basic advantages for retrieval (vs documents)

Passages

- Some basic advantages for retrieval (vs documents)
  - Documents vary in

Passages

- Some basic advantages for retrieval (vs documents)
  - Documents vary in
    - Length,
    - Topic term density,
    - Etc
      - across type

Passages

- Some basic advantages for retrieval (vs documents)
  - Documents vary in
    - Length,
    - Topic term density,
    - Etc
      - across type
  - Passages can be less variable
    - Effectively normalizing for length

What Makes a Passage?

- Sources of passage information

What Makes a Passage?

- Sources of passage information
  - Manual:
    - Existing markup

What Makes a Passage?

- Sources of passage information
  - Manual:
    - Existing markup
      - E.g., Sections, Paragraphs
      - Issues: ?

What Makes a Passage?

- Sources of passage information
  - Manual:
    - Existing markup
      - E.g., Sections, Paragraphs
      - Issues: ?
        - Still highly variable:
          - Wikipedia vs Newswire

What Makes a Passage?

- Sources of passage information
  - Manual:
    - Existing markup
      - E.g., Sections, Paragraphs
      - Issues: ?
        - Still highly variable:
          - Wikipedia vs Newswire
        - Potentially ambiguous:
          - blank lines separate .....

What Makes a Passage?

- Sources of passage information
  - Manual:
    - Existing markup
      - E.g., Sections, Paragraphs
      - Issues: ?
        - Still highly variable:
          - Wikipedia vs Newswire
        - Potentially ambiguous:
          - blank lines separate .....
        - Not always available

What Makes a Passage?

- Automatic:

What Makes a Passage?

- Automatic:
  - Semantically motivated document segmentation
    - Linguistic content
    - Lexical patterns and relations

What Makes a Passage?

- Automatic:
  - Semantically motivated document segmentation
    - Linguistic content
    - Lexical patterns and relations
  - Fixed length units:
    - In words/chars or sentences/paragraphs
    - Overlapping?
    - Can be determined empirically
- All experiments use Zettair retrieval engine

Coreference Chains

- Coreference:
  - NPs that refer to same entity
    - Create an equivalence class

Coreference Chains

- Coreference:
  - NPs that refer to same entity
    - Create an equivalence class
  - Chains of coreference suggest entity-based coherence

Coreference Chains

- Coreference:
  - NPs that refer to same entity
    - Create an equivalence class
  - Chains of coreference suggest entity-based coherence
- Passage:
  - All sentences spanned by a coreference chain
  - Can create overlapping passages
  - Built with cluster-based ranking with own coref. system
    - System has F-measure of 54.5%

Coreference Chains

- Coreference:
  - NPs that refer to same entity
    - Create an equivalence class
  - Chains of coreference suggest entity-based coherence
- Passage:
  - All sentences spanned by a coreference chain
  - Can create overlapping passages
  - Built with cluster-based ranking with own coref. System
    - System has F-measure of 54.5%

1. [Jim McClements en Susan Sandvig-Shobe]$_i$ hebben een onrechtmatig argument gebruikt.
2. [De Nederlandse scheidsrechter]$_j$ [Jacques de Koning]$_j$ bevestigt dit.
3. [Kuipers]$_k$ versloeg zondag in een rechtstreeks duel [Shani Davis]$_m$.
4. Toch werd [hij]$_k$ in de rangschikking achter [de Amerikaan]$_m$ geklasseerd.
5. [De twee hoofdarbiters]$_i$ verklaarden dat [Kuipers']$_k$ voorste schaats niet op de grond stond.

Cluster i (1,5): [Jim McClements en Susan Sandvig-Shobe]
[De twee hoofdarbiters]

Cluster j (2): [De Nederlandse scheidsrechter]
[Jacques de Koning]

Cluster k (3-5): [Kuipers] [hij] [Kuipers']

Cluster m (3,4): [Shani Davis] [de Amerikaan]

TextTiling (Hearst)

- Automatic topic, sub-topic segmentation

TextTiling (Hearst)

- Automatic topic, sub-topic segmentation
  - Computes similarity between neighboring text blocks
    - Based on weighted cosine similarity

TextTiling (Hearst)

- Automatic topic, sub-topic segmentation
  - Computes similarity between neighboring text blocks
    - Based on weighted cosine similarity
  - Compares similarity values
    - Hypothesizes topic shift at dips b/t peaks in similarity

TextTiling (Hearst)

- Automatic topic, sub-topic segmentation
  - Computes similarity between neighboring text blocks
    - Based on weighted cosine similarity
  - Compares similarity values
    - Hypothesizes topic shift at dips b/t peaks in similarity
  - Produces linear topic segmentation

TextTiling (Hearst)

- Automatic topic, sub-topic segmentation
  - Computes similarity between neighboring text blocks
    - Based on weighted cosine similarity
  - Compares similarity values
    - Hypothesizes topic shift at dips b/t peaks in similarity
  - Produces linear topic segmentation
  - Existing implementations

Window-based Segmentation

- Fixed width windows:
  - Based on words? Characters? Sentences?

Window-based Segmentation

- Fixed width windows:
  - Based on words? Characters? Sentences?
    - Sentences required for downstream deep processing

Window-based Segmentation

- Fixed width windows:
  - Based on words? Characters? Sentences?
    - Sentences required for downstream deep processing
  - Overlap? No overlap?

Window-based Segmentation

- Fixed width windows:
  - Based on words? Characters? Sentences?
    - Sentences required for downstream deep processing
  - Overlap? No overlap?
    - No overlap is simple, but
      - Not guaranteed to line up with natural boundaries
        - Including document boundaries

Evaluation

- Indexing and retrieval in Zettair system
  - CLEF Dutch QA track
- Computes
  - Lenient MRR measure

Evaluation

- Indexing and retrieval in Zettair system
  - CLEF Dutch QA track
- Computes
  - Lenient MRR measure
    - Too few participants to assume pooling exhaustive
  - Redundancy: Average # relevant passage per query

Evaluation

- Indexing and retrieval in Zettair system
  - CLEF Dutch QA track
- Computes
  - Lenient MRR measure
    - Too few participants to assume pooling exhaustive
  - Redundancy: Average # relevant passage per query
  - Coverage: Proportion of Qs w/at least one relpass

Evaluation

- Indexing and retrieval in Zettair system
  - CLEF Dutch QA track
- Computes
  - Lenient MRR measure
    - Too few participants to assume pooling exhaustive
  - Redundancy: Average # relevant passage per query
  - Coverage: Proportion of Qs w/at least one relpass
  - MAP

Evaluation

- Indexing and retrieval in Zettair system
  - CLEF Dutch QA track
- Computes
  - Lenient MRR measure
    - Too few participants to assume pooling exhaustive
  - Redundancy: Average # relevant passage per query
  - Coverage: Proportion of Qs w/at least one relpass
  - MAP
- Focus on MRR for prediction of end-to-end QA

Baselines

- Existing markup:
  - Documents, paragraphs, sentences

Baselines

- Existing markup:
  - Documents, paragraphs, sentences
- MRR-IR; MRR-QA (top 5); CLEF: end-to-end score

Baselines

- Existing markup:
  - Documents, paragraphs, sentences
- MRR-IR; MRR-QA (top 5); CLEF: end-to-end score
- Surprisingly good sentence results in top-5 and CLEF
  - Sensitive to exact retrieval weighting

	#sent	cov	red	*MRR* IR	*MRR* QA	CLEF
sent	16,737	0.784	2.95	0.490	**0.487**	**0.430**
par	80,046	0.842	4.17	0.565	0.483	0.416
doc	618,865	0.877	6.13	0.666	0.457	0.387

Semantic Passages

- Contrast:
  - Sentence/coref: Sentences in coref. chains

Semantic Passages

- Contrast:
  - Sentence/coref: Sentences in coref. chains → too long

Semantic Passages

- Contrast:
  - Sentence/coref: Sentences in coref. chains → too long
    - Bounded length
  - Paragraphs and coref chains (bounded)

Semantic Passages

- Contrast:
  - Sentence/coref: Sentences in coref. chains → too long
    - Bounded length
  - Paragraphs and coref chains (bounded)
  - TextTiling (CPAN)

Semantic Passages

- Contrast:
  - Sentence/coref: Sentences in coref. chains → too long
    - Bounded length
  - Paragraphs and coref chains (bounded)
  - TextTiling (CPAN) – Best : beats baseline

	#sent	MRR IR	MRR QA	CLEF
sent/coref	490,968	**0.604**	0.469	0.405
sent/coref (200-1000)	76,865	0.535	0.462	0.395
par+coref (200-1000)	82,378	0.560	**0.493**	0.426
par+coref (200-400)	67,580	0.555	**0.489**	0.422
TextTiling	107,879	**0.586**	△ **0.503**	**0.434**

Fixed Size Windows

- Different lengths: non-overlapping
- 2-, 4-sentence units improve over semantic units

	#sent	MRR IR	MRR QA	CLEF
2 sentences	33468	0.545	△ **0.506**	**0.443**
3 sentences	50190	0.554	**0.504**	**0.436**
4 sentences	66800	**0.581**	△ **0.512**	**0.447**
5 sentences	83575	**0.588**	**0.493**	0.422
6 sentences	100110	**0.583**	**0.489**	0.423

Sliding Windows

- Fixed length windows, overlapping
- Best MRR-QA values
  - Small units with overlap
  - Other settings weaker

	#sent	MRR IR	MRR QA	CLEF
2 sent (sliding)	29095	0.548	△ **0.516**	**0.456**
3 sent (sliding)	36415	0.549	**0.484**	0.411
4 sent (sliding)	41565	0.546	0.476	0.409
5 sent (sliding)	45737	0.534	0.465	0.403
6 sent (sliding)	49091	0.528	0.454	0.390

Observations

- Competing retrieval demands:
  - IR performance
    - vs
  - QA performance

Observations

- Competing retrieval demands:
  - IR performance
    - vs
  - QA performance
- MRR at 5 favors:

Observations

- Competing retrieval demands:
  - IR performance
    - vs
  - QA performance
- MRR at 5 favors:
  - Small, fixed width units
    - Advantageous for downstream processing too

Observations

- Competing retrieval demands:
  - IR performance
    - vs
  - QA performance
- MRR at 5 favors:
  - Small, fixed width units
    - Advantageous for downstream processing too
  - Any benefit of more sophisticated segments
    - Outweighed by increased processing

Reranking with Deep Processing

- Passage Reranking for Question Answering Using Syntactic Structures and Answer Types
  - Atkolga et al, 2011
- Reranking of retrieved passages
  - Integrates
    - Syntactic alignment
    - Answer type
    - Named Entity information

Motivation

- Issues in shallow passage approaches:
  - From Tellex et al.

Motivation

- Issues in shallow passage approaches:
  - From Tellex et al.
    - Retrieval match admits many possible answers
      - Need answer type to restrict

Motivation

- Issues in shallow passage approaches:
  - From Tellex et al.
    - Retrieval match admits many possible answers
      - Need answer type to restrict
    - Question implies particular relations
      - Use syntax to ensure

Motivation

- Issues in shallow passage approaches:
  - From Tellex et al.
    - Retrieval match admits many possible answers
      - Need answer type to restrict
    - Question implies particular relations
      - Use syntax to ensure
  - Joint strategy required
    - Checking syntactic parallelism when no answer, useless
- Current approach incorporates all (plus NER)

Baseline Retrieval

- Bag-of-Words unigram retrieval (BOW)

Baseline Retrieval

- Bag-of-Words unigram retrieval (BOW)

- Question analysis: QuAn
  - ngram retrieval, reformulation

Baseline Retrieval

- Bag-of-Words unigram retrieval (BOW)
- Question analysis: QuAn
  - ngram retrieval, reformulation
- Question analysis + Wordnet: QuAn-Wnet
  - Adds 10 synonyms of ngrams in QuAn

Baseline Retrieval

- Bag-of-Words unigram retrieval (BOW)
- Question analysis: QuAn
  - ngram retrieval, reformulation
- Question analysis + Wordnet: QuAn-Wnet
  - Adds 10 synonyms of ngrams in QuAn
- Best performance: QuAn-Wnet (baseline)

Dependency Information

- Assume dependency parses of questions, passages
  - Passage = sentence
- Extract undirected dependency paths b/t words

Dependency Information

- Assume dependency parses of questions, passages
  - Passage = sentence
- Extract undirected dependency paths b/t words
- Find path pairs between words $(q_k,a_l),(q_r,a_s)$
  - Where q/a words ‘match’
    - Word match if a) same root or b) synonyms

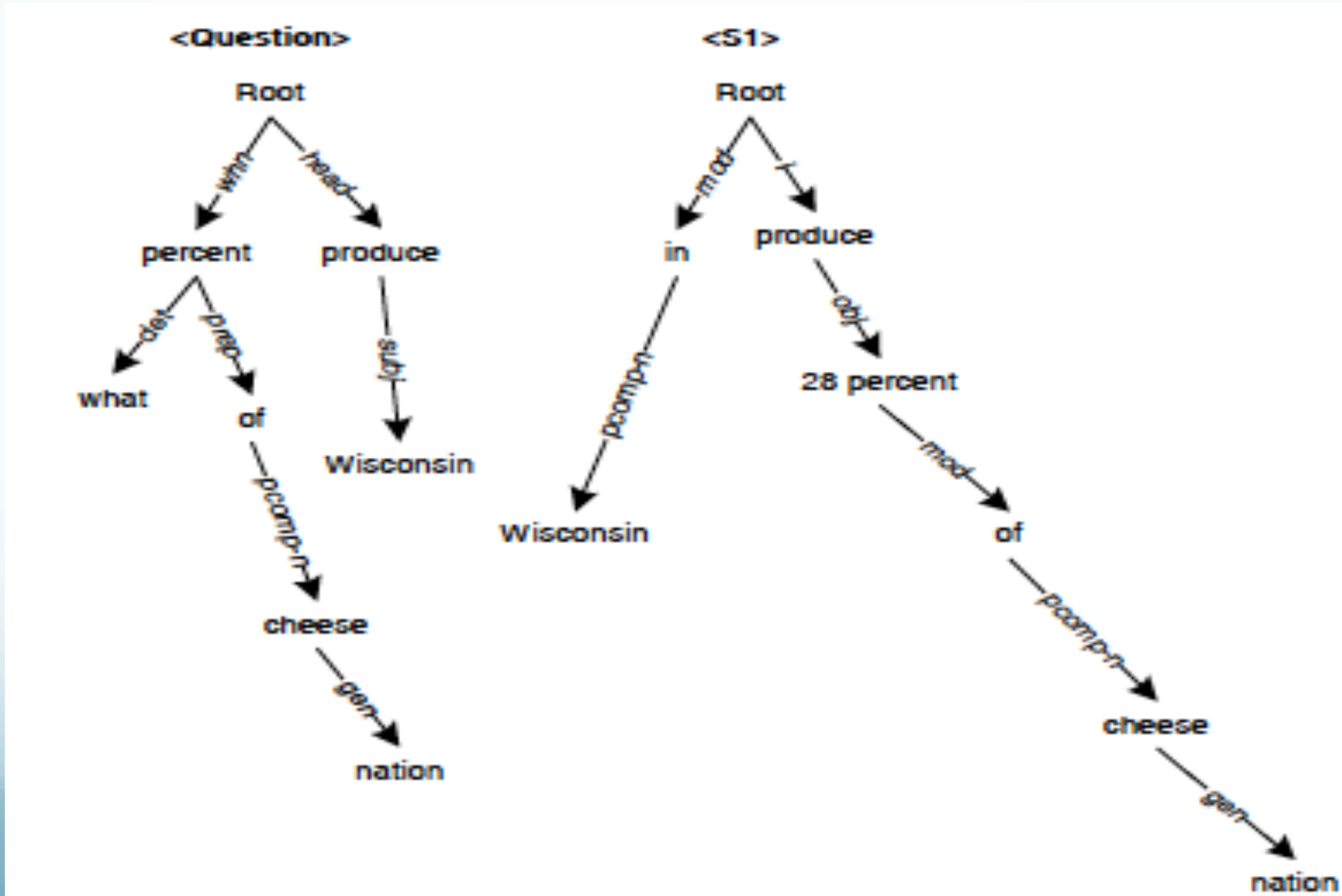
Dependency Information

- Assume dependency parses of questions, passages
  - Passage = sentence
- Extract undirected dependency paths b/t words
- Find path pairs between words $(q_k,a_l),(q_r,a_s)$
  - Where q/a words 'match'
    - Word match if a) same root or b) synonyms
    - Later: require one pair to be question word/Answer term
- Train path 'translation pair' probabilities

Dependency Information

- Assume dependency parses of questions, passages
  - Passage = sentence
- Extract undirected dependency paths b/t words
- Find path pairs between words $(q_k,a_l),(q_r,a_s)$
  - Where q/a words 'match'
    - Word match if a) same root or b) synonyms
    - Later: require one pair to be question word/Answer term
- Train path 'translation pair' probabilities
  - Use true Q/A pairs, $<path_q,path_a>$
  - GIZA++, IBM model 1
    - Yields $Pr(label_a,label_q)$

Dependency Path Similarity



Dependency Path Similarity

Figure 2. Dependency trees for the sample question and sentence S1 in Figure 1 generated by Minipar. Some nodes are omitted due to lack of space.

Question:

Path_ID	Node1	Path	Node2
<P_{Q1}>	Wisconsin	*<subj>*	produce
<P_{Q2}>	produce	*<head, whn, prep, pcomp-n>*	cheese
<P_{Q3}>	nation	<gen>	cheese

S1:

Path_ID	Node1	Path	Node2
<P_{S1}>	Wisconsin	*<pcomp-n, mod, i>*	produce
<P_{S2}>	produce	*<obj, mod, pcomp-n>*	cheese
<P_{S3}>	nation	*<gen>*	cheese

Similarity

- Dependency path matching

Similarity

- Dependency path matching
  - Some paths match exactly
  - Many paths have partial overlap or differ due to question/declarative contrasts

Similarity

- Dependency path matching
  - Some paths match exactly
  - Many paths have partial overlap or differ due to question/declarative contrasts

- Approaches have employed
  - Exact match
  - Fuzzy match
  - Both can improve over baseline retrieval, fuzzy more

Dependency Path Similarity

- Cui et al scoring
- Sum over all possible paths in a QA candidate pair

Dependency Path Similarity

- Cui et al scoring
- Sum over all possible paths in a QA candidate pair

$$\sum_{path_q, path_a \in Paths} scorePair(path_q, path_a)$$

Dependency Path Similarity

- Cui et al scoring
- Sum over all possible paths in a QA candidate pair

$$\sum_{path_q, path_a \in Paths} scorePair(path_q, path_a)$$

$$\frac{1}{|path_a|} \prod_{label_{a_j}} \sum_{label_{q_t}} \Pr(label_{a_j} \mid label_{q_t})$$

Dependency Path Similarity

- Atype-DP
- Restrict first q,a word pair to Qword, ACand
  - Where Acand has correct answer type by NER

Dependency Path Similarity

- Atype-DP
- Restrict first q,a word pair to Qword, ACand
  - Where Acand has correct answer type by NER
- Sum over all possible paths in a QA candidate pair
  - with best answer candidate

Dependency Path Similarity

- Atype-DP
- Restrict first q,a word pair to Qword, ACand
  - Where Acand has correct answer type by NER
- Sum over all possible paths in a QA candidate pair
  - with best answer candidate

$$\max_i \sum_{path_q, path_a \in Paths_{ACand_i}} scorePair(path_q, path_a)$$

Comparisons

- Atype-DP-IP
  - Interpolates DP score with original retrieval score

Comparisons

- Atype-DP-IP
  - Interpolates DP score with original retrieval score

- QuAn-Elim:
  - Acts a passage answer-type filter
  - Excludes any passage w/o correct answer type

Results

- Atype-DP-IP best

Table 2. Evaluation of Reranking Techniques. All results are averages from the testing datasets TREC 2000 and TREC 2001, evaluated on the top 100 retrieved passages.

Model	*MRR@1*	*MRR@5*	*MRR@10*	*MRR@20*	*MRR@50*	*MRR@100*
Q-BOW	0.168	0.266	0.286	0.293	0.299	0.301
QuAn-Wnet	0.193	0.289	0.308	0.319	0.324	0.325
Cui	0.202	0.307	0.325	0.335	0.339	0.341
Atype-DP	0.148	0.24	0.26	0.273	0.279	0.28
Atype-DP-IP	**0.261***	**0.363***	**0.38***	**0.389***	**0.393***	**0.394***
% Improvement over Cui	**+29.2**	+18.24	+16.9	+16.12	+15.9	+15.54
% Improvement over QuAn-Wnet	**+35.2**	+25.6	+23.4	+21.9	+21.3	+ 21.2

Results

- Atype-DP-IP best
  - Raw dependency:'brittle'; NE failure backs off to IP

Table 2. Evaluation of Reranking Techniques. All results are averages from the testing datasets TREC 2000 and TREC 2001, evaluated on the top 100 retrieved passages.

Model	*MRR@1*	*MRR@5*	*MRR@10*	*MRR@20*	*MRR@50*	*MRR@100*
Q-BOW	0.168	0.266	0.286	0.293	0.299	0.301
QuAn-Wnet	0.193	0.289	0.308	0.319	0.324	0.325
Cui	0.202	0.307	0.325	0.335	0.339	0.341
Atype-DP	0.148	0.24	0.26	0.273	0.279	0.28
Atype-DP-IP	**0.261***	**0.363***	**0.38***	**0.389***	**0.393***	**0.394***
% Improvement over Cui	**+29.2**	+18.24	+16.9	+16.12	+15.9	+15.54
% Improvement over QuAn-Wnet	**+35.2**	+25.6	+23.4	+21.9	+21.3	+ 21.2

Results

- Atype-DP-IP best
  - Raw dependency:'brittle'; NE failure backs off to IP
- QuAn-Elim: NOT significantly worse

Table 2. Evaluation of Reranking Techniques. All results are averages from the testing datasets TREC 2000 and TREC 2001, evaluated on the top 100 retrieved passages.

Model	*MRR@1*	*MRR@5*	*MRR@10*	*MRR@20*	*MRR@50*	*MRR@100*
Q-BOW	0.168	0.266	0.286	0.293	0.299	0.301
QuAn-Wnet	0.193	0.289	0.308	0.319	0.324	0.325
Cui	0.202	0.307	0.325	0.335	0.339	0.341
Atype-DP	0.148	0.24	0.26	0.273	0.279	0.28
Atype-DP-IP	**0.261***	**0.363***	**0.38***	**0.389***	**0.393***	**0.394***
% Improvement over Cui	**+29.2**	+18.24	+16.9	+16.12	+15.9	+15.54
% Improvement over QuAn-Wnet	**+35.2**	+25.6	+23.4	+21.9	+21.3	+ 21.2

Learning Passage Ranking

- Alternative to heuristic similarity measures
- Identify candidate features
- Allow learning algorithm to select

Learning Passage Ranking

- Alternative to heuristic similarity measures
- Identify candidate features
- Allow learning algorithm to select
- Learning and ranking:
  - Employ general classifiers
    - Use score to rank (e.g., SVM, Logistic Regression)

Learning Passage Ranking

- Alternative to heuristic similarity measures
- Identify candidate features
- Allow learning algorithm to select
- Learning and ranking:
  - Employ general classifiers
    - Use score to rank (e.g., SVM, Logistic Regression)
  - Employ explicit rank learner
    - E.g. RankBoost

Shallow Features & Ranking

- Is Question Answering an Acquired Skill?
  - Ramakrishnan et al, 2004
- Full QA system described
  - Shallow processing techniques
  - Integration of Off-the-shelf components
  - Focus on rule-learning vs hand-crafting
  - Perspective: questions as noisy SQL queries

Architecture

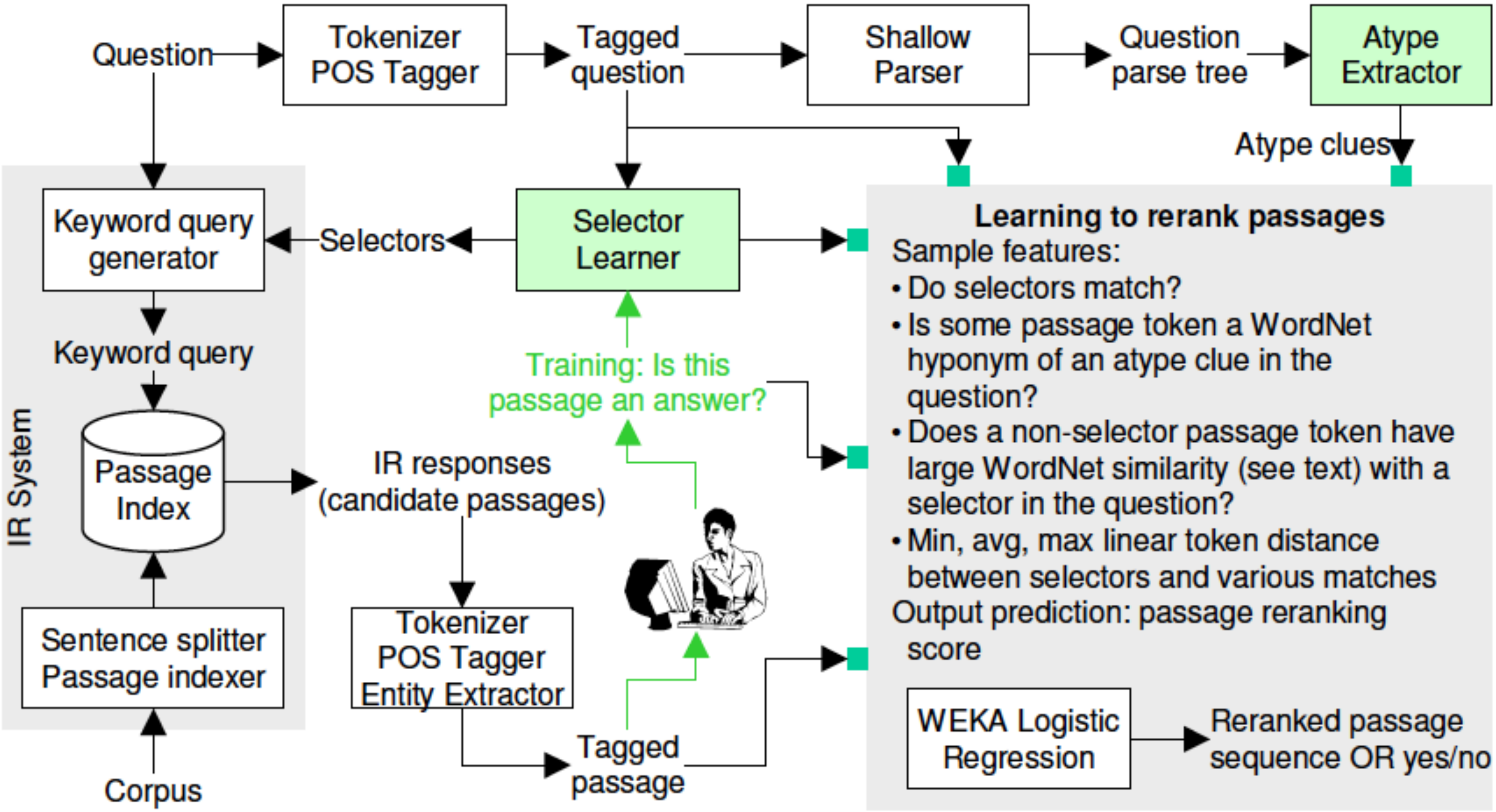


Figure 2: Overall architecture of our trainable QA system.

Basic Processing

- Initial retrieval results:
  - IR 'documents':
    - 3 sentence windows (Tellex et al)
  - Indexed in Lucene
  - Retrieved based on reformulated query

Basic Processing

- Initial retrieval results:
  - IR ‘documents’:
    - 3 sentence windows (Tellex et al)
  - Indexed in Lucene
  - Retrieved based on reformulated query
- Question-type classification
  - Based on shallow parsing
  - Synsets or surface patterns

Selectors

- Intuition:
  - ‘Where’ clause in an SQL query – selectors

Selectors

- Intuition:
  - ‘Where’ clause in an SQL query – selectors
  - Portion(s) of query highly likely to appear in answer
- Train system to recognize these terms
  - Best keywords for query
  - *Tokyo is the capital of which country?*
    - Answer probably includes.....

Selectors

- Intuition:
  - ‘Where’ clause in an SQL query – selectors
  - Portion(s) of query highly likely to appear in answer
- Train system to recognize these terms
  - Best keywords for query
  - *Tokyo is the capital of which country?*
    - Answer probably includes.....
      - Tokyo+++
      - Capital+
      - Country?

Selector Recognition

- Local features from query:
  - POS of word
  - POS of previous/following word(s), in window
  - Capitalized?

Selector Recognition

- Local features from query:
  - POS of word
  - POS of previous/following word(s), in window
  - Capitalized?
- Global features of word:
  - Stopword?
  - IDF of word
  - Number of word senses
  - Average number of words per sense

Selector Recognition

- Local features from query:
  - POS of word
  - POS of previous/following word(s), in window
  - Capitalized?
- Global features of word:
  - Stopword?
  - IDF of word
  - Number of word senses
  - Average number of words per sense
    - Measures of word specificity/ambiguity

Selector Recognition

- Local features from query:
  - POS of word
  - POS of previous/following word(s), in window
  - Capitalized?
- Global features of word:
  - Stopword?
  - IDF of word
  - Number of word senses
  - Average number of words per sense
    - Measures of word specificity/ambiguity
- Train Decision Tree classifier on gold answers: +/-S

Passage Ranking

- For question q and passage r, in a good passage:

Passage Ranking

- For question q and passage r, in a good passage:
  - All selectors in q appear in r

Passage Ranking

- For question q and passage r, in a good passage:
  - All selectors in q appear in r
  - r has answer zone A w/o selectors

Passage Ranking

- For question q and passage r, in a good passage:
  - All selectors in q appear in r
  - r has answer zone A w/o selectors
  - Distances b/t selectors and answer zone A are small

Passage Ranking

- For question q and passage r, in a good passage:
  - All selectors in q appear in r
  - r has answer zone A w/o selectors
  - Distances b/t selectors and answer zone A are small
  - A has high similarity with question type

Passage Ranking

- For question q and passage r, in a good passage:
  - All selectors in q appear in r
  - r has answer zone A w/o selectors
  - Distances b/t selectors and answer zone A are small
  - A has high similarity with question type
  - Relationship b/t Qtype, A's POS and NE tag (if any)

Passage Ranking Features

- Find candidate answer zone A* as follows for (q.r)
  - Remove all matching q selectors in r
  - For each word (or compound in r) A
    - Compute Hyperpath distance b/t Qtype & A
      - Where HD is Jaccard overlap between hypernyms of Qtype & A

Passage Ranking Features

- Find candidate answer zone A* as follows for (q.r)
  - Remove all matching q selectors in r
  - For each word (or compound in r) A
    - Compute Hyperpath distance b/t Qtype & A
      - Where HD is Jaccard overlap between hypernyms of Qtype & A
- Compute L as set of distances from selectors to A*
- Feature vector:

Passage Ranking Features

- Find candidate answer zone A* as follows for (q.r)
  - Remove all matching q selectors in r
  - For each word (or compound in r) A
    - Compute Hyperpath distance b/t Qtype & A
      - Where HD is Jaccard overlap between hypernyms of Qtype & A
- Compute L as set of distances from selectors to A*
- Feature vector:
  - IR passage rank; HD score; max, mean, min of L

Passage Ranking Features

- Find candidate answer zone A* as follows for (q.r)
  - Remove all matching q selectors in r
  - For each word (or compound in r) A
    - Compute Hyperpath distance b/t Qtype & A
      - Where HD is Jaccard overlap between hypernyms of Qtype & A
- Compute L as set of distances from selectors to A*
- Feature vector:
  - IR passage rank; HD score; max, mean, min of L
  - POS tag of A*; NE tag of A*; Qwords in q

Passage Ranking

- Train logistic regression classifier
  - Positive example:

Passage Ranking

- Train logistic regression classifier
  - Positive example: question + passage with answer
  - Negative example:

Passage Ranking

- Train logistic regression classifier
  - Positive example: question + passage with answer
  - Negative example: question w/any other passage
- Classification:
  - Hard decision: 80% accurate, but

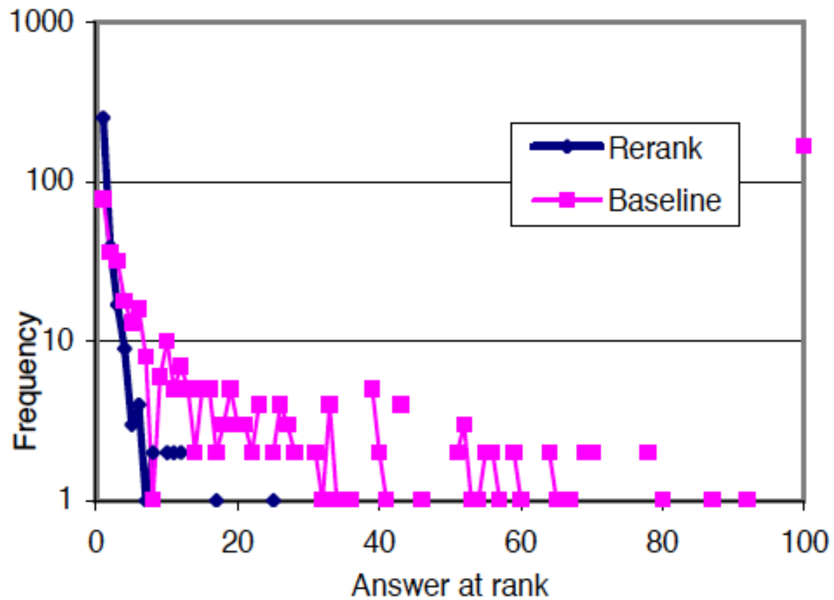
Passage Ranking

- Train logistic regression classifier
  - Positive example: question + passage with answer
  - Negative example: question w/any other passage
- Classification:
  - Hard decision: 80% accurate, but
    - Skewed, most cases negative: poor recall

Passage Ranking

- Train logistic regression classifier
  - Positive example: question + passage with answer
  - Negative example: question w/any other passage
- Classification:
  - Hard decision: 80% accurate, but
    - Skewed, most cases negative: poor recall
- Use regression scores directly to rank

Passage Ranking



igure 9: Reranking significantly improves the rank
f correct passages. The x-axis is the rank at which

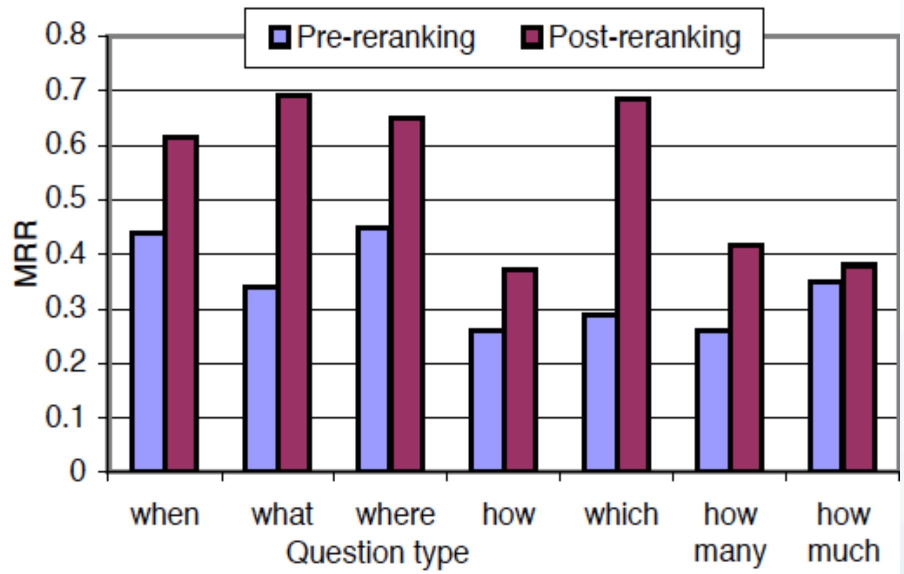


Figure 12: Sample MRR improvement via reranking separated into question categories.