DELIVERABLE 4: FINAL QA SYSTEM
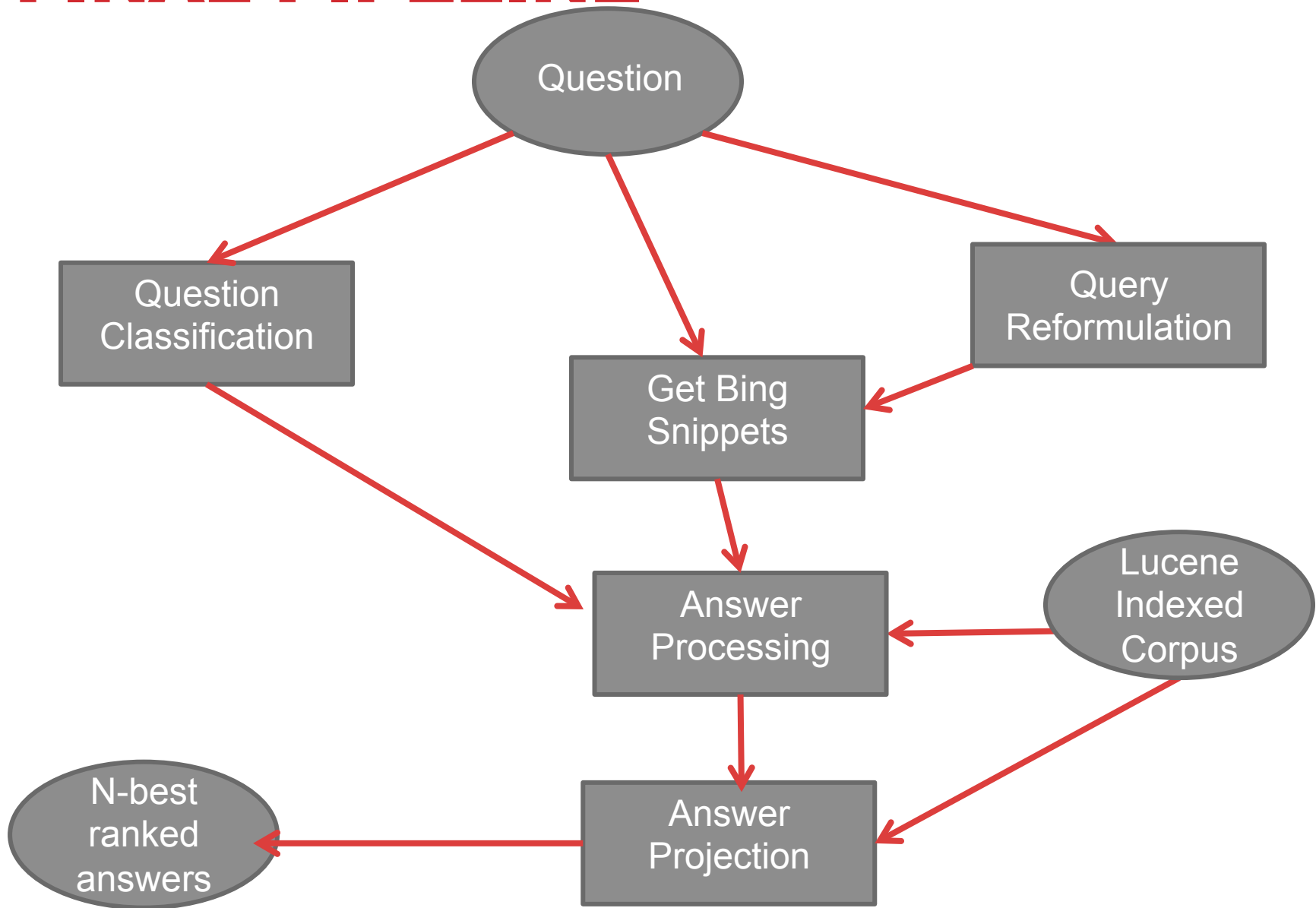
JOHN GILMER

MICHAEL FOSTER

ADAM LEDYARD

WEb Based Answer eXtraction

OVERVIEW

- ➢ **Pipeline**
- ➢ **Redundancy Approach**
- ➢ **Question Classification**
- ➢ **Question Reformulation**
- ➢ **Answer Processing**
- ➢ **Results**
- ➢ **Other Strategies**
- ➢ **References**

FINAL PIPELINE

Question → Question Classification

Question → Get Bing Snippets

Question → Query Reformulation

Query Reformulation → Get Bing Snippets

Question Classification → Answer Processing

Get Bing Snippets → Answer Processing

Lucene Indexed Corpus → Answer Processing

Answer Processing → Answer Projection

Lucene Indexed Corpus → Answer Projection

Answer Projection → N-best ranked answers

REDUNDANCY APPROACH

Ngram Generation:

Snippets from the query results are broken up into unigrams, bigrams, trigrams, and tetragrams

Ngram Filtering:

Ngrams are removed from consideration if the begin with or end with a stopword, certain how-type answer require numerals, NERs (first and last word is capitalized) are only kept with 'thing' and 'location' type answer candidate

REDUNDANCY APPROACH

Ngram Combining:

      Unigram Counts are added to bigram, trigram and tetragram counts and reranked

REDUNDANCY APPROACH

Ngram Scoring:

$$S{\downarrow}NGRAM = CC{\downarrow}NGRAM * \sum u \in NGRAM{\uparrow}\blacksquare \log(IDF(u))/|NGRAM|$$

$$S = Score$$

$$CC = Combined\ Count$$

$$u = unigram$$

$$|NGRAM| = number\ of\ unigrams$$

IDF calculations were done with the Lucene API on the AQUAINT corpus.

QUESTION CLASSIFICATION

➢ **Used Roth and Li's Question Classification Taxonomy**

➢ **Trained a MaxEnt Classifier on the 5500 training set**

   ➢ Features used:

      ➢ Unigrams

      ➢ POS tag unigrams

      ➢ NP chunks found by nltk trained chunker using BIO tagged CoNLL 2000 corpus (WSJ text)

      ➢ Semantic feature based on Roth and Li's semantic lists

   ➢ Results: 84% accuracy based on TREC 10 labeled test data.

QUESTION CLASSIFICATION

Output:

Each Question was classified and a confidence score was attached to the Question Type that was used later in answer processing.

Answer Processing : Filter Method

Based on the Question type Ngram answer candidates were either boosted or removed from consideration.

Open Class Answer Types → Boost scores

Ex. Cities, Names

Closed Class Answer Types → Delete

Ex. Mountains, Instruments, Numbers

QUESTION REFORMULATION

➢ **Produced 2 similar queries for each question.**

➢ **Query 1: Question text with topic inserted:**

  ➢ Replace pronouns with topic if present.

  ➢ Check for partial topic text and fill in as needed.

  ➢ Otherwise, insert topic in front of the question.

  ➢ Remove stop words and punctuation.

➢ **Query 2: Same as query 1 but with topic in double quotes.**

  ➢ Answers candidates that came from queries with quotes later receive a boost to their score.

➢ **Didn't use exact query reformulation due to throttling and time limitations**

ANSWER PROCESSING

1. **Answer Processing by Question Type.**

2. **Boost Ngram counts from the first 10 snippets from Bing.**

3. **Check AQUAINT corpus**

    1. Collect all proper noun phrases from topic
    2. Remove answer from candidate answers if Lucene cannot find a document that contains both a proper noun from the topic and the full ngram answer by using Lucene's Phrase Query and Boolean Query classes.

RESULTS

TREC 2006

	Strict	Lenient
100 chars	0.106	0.239
250 chars	0.107	0.239

TREC 2007

	Strict	Lenient
100 chars	0.124	0.259
250 chars	0.124	0.259

OTHER STRATEGIES

FRAMENET AND RERANKING

➢ **Tried developing a Re-ranker based on Ravichandran et al. using modified feature selection based on what we could extract from our system.**

➢ **Modified features:**

  ➢ Used overlapping frames in FrameNet instead of patterns firing

    ➢ Found all frames one relation away in the hierarchies in common with question and all snippet content-full words

  ➢ Number of words in the topic contained in snippet

  ➢ Number of words in the query contained in the snippet

  ➢ Binary feature if in Bing's Top Ten results

  ➢ Binary feature if snippet is result from exact match query

  ➢ Answer Type with value equal to 3 * confidence score

UNIQUE ANSWER CANDIDATE RERANKING

➢ **Borrowed concept from Tristan, Stefan & Chris' group.**

➢ **Reranked answer candidates based on the number of repeated terms.**

➢ **Down weighted answer candidate score as follows:**

    ➢ $s = s * c \wedge n$

    ➢ Where:

        ➢ s = Answer candidate score.

        ➢ c = A constant between 0 and 1.

        ➢ n = the number of terms in the answer that were found in other answers

➢ **No boost to MRR from this approach.**