# A Non-factoid Question Answering System

# for Tweet Contextualization

A thesis

Submitted to the faculty of the Graduate school

of the University of Minnesota

by

Swapnil Atmaram Nawale

In partial fulfillment of the requirements

for the degree of

Master of Science

Dr. Donald B. Crouch

August 2013

## Acknowledgements

I would like to express my sincere gratitude to both Dr. Donald Crouch and Dr. Carolyn Crouch for their continuous support and guidance in this research work. I would also thank Dr. Marshall Hampton for guiding me in one of the best math course on Bioinformatics and for being a part of thesis committee. I would also like to acknowledge Dr. Ted Pedersen for a wonderful course on Natural Language Processing, which directly helped me in delivering this thesis. Furthermore, I am grateful to Lori Lucia and Clare Ford for their help throughout two years of my graduate degree.

Last but not the least, I would like to thank my parents Atmaram and Asha Nawale for their support, kindness and love in all my endeavors without which I would have never been able to reach this stage of my life.

## Abstract

Information Retrieval (IR) is a field that deals with the storage and retrieval of information from a large collection of documents. A document consists primarily of text, for example, a webpage or a news article. IR attempts to satisfy the information need of the user. Traditionally, the user enters a natural language query, and documents containing information about that query are returned by the system. But in many cases, the user may be interested in specific and concise pieces of information rather than an entire document. One such scenario occurs in the field of Question Answering (QA). In QA, the user enters a natural language question and QA systems come up with a concise answer to the user's question. The question can be factoid or non-factoid. Factoid questions have simple facts as answers, and these facts are retrieved from a single document, whereas non-factoid questions typically have as answers longer pieces of readable information which may come from single or multiple documents.

This thesis describes a non-factoid QA system developed for a retrieval task known as Tweet Contextualization [5]. Our QA system for the Tweet Contextualization task takes tweets from microblogging website *Twitter* as an input and provides an answer to the question: *What is this tweet about?*, I.e., it provides the context for the tweet. This context is in the form of a maximum 500 word summary and is extracted from the recent, cleaned Wikipedia dump [2]. We use Indri [15] as a primary retrieval tool for our QA system. We also describe our approach for generating context summaries by considering n-gram overlap between tweets and sentences from the Wikipedia corpus. The top-ranked results achieved by our QA system for the INEX [2] 2012 and 2013 Tweet Contextualization tracks are also included.

# Table of Contents

## List of Tables

## List of Figures

# 1. Introduction

Information Retrieval (IR) is a field that deals with the storage and retrieval of information from a large collection of documents. A document consists primarily of text, for example, a webpage or a news article. IR attempts to satisfy the information need of the user. Traditionally, the user enters a natural language query and documents containing information about that query are returned by the system.

But in many cases, the user may be interested in specific and concise pieces of information rather than an entire document. One such scenario occurs in the field of Question Answering (QA). In QA, the user enters a natural language question and QA systems come up with a concise answer to the user's question. The question can be factoid or non-factoid. Factoid questions have simple facts as answer and these facts are retrieved from a single document, e.g., *Where was Julius Caesar born?* Non-factoid questions typically have answers as longer pieces of readable information, which might come from a single or multiple documents, for example, *What were the consequences of World War II?*

The objective of this research is to provide a robust and efficient non-factoid QA system for the Tweet Contextualization track [5] of the Initiative for the Evaluation of XML Retrieval (INEX) forum [2]. The goal of INEX is to enhance XML retrieval. INEX provides the basic infrastructure for various tracks and a platform for the evaluation of those tracks. Tweet Contextualization is one such track in INEX 2013. For the Tweet Contextualization track, participants are provided with public tweets from the micro-blogging website *Twitter*. Participants must devise a QA system which will answer the question: *What is this tweet about?* I.e., they must provide the context for each given tweet. This context is in the form of a maximum 500 word summary to be extracted from a recent, cleaned Wikipedia dump [2] with documents in structured XML format. The maximum length of a tweet is 140 characters. Because of their typical shortness, tweets

usually carry less information within themselves. This makes the task of contextualization difficult.

An important step for any QA system is indexing. Indexing is the process that facilitates the storage and retrieval of the documents. We use the Indri text search engine [15] to index the Wikipedia documents for our QA system. Indri combines the approaches of Language Modeling (LM) with inference networks for indexing documents. LM is a probabilistic approach used in document indexing. This approach creates a language model for each document. Documents are then retrieved based on the probabilities of producing query terms from the corresponding language models of these documents [14]. The inference network approach assumes that a query is made up of concepts. A concept can be represented by query terms, phrases or other complex entities present in queries. Inference networks assume a document is relevant to a query only when it contains concepts present in the query [15].

The results of the Tweet Contextualization track are evaluated based upon two metrics, namely, (1) informativeness of the context summary generated by QA system, and (2) overall readability of the context summary generated by QA system.

The details of the INEX Tweet Contextualization track along with information about its two evaluation metrics are described in Chapter 2. The strategies used by our QA system for achieving higher informativeness and readability in the context summaries are discussed in Chapter 3. Chapter 4 describes the experiments performed for the Tweet Contextualization track and evaluation of those experiments. Conclusions and future scope are presented in Chapter 5.

## 2. Background

This chapter gives an overview of INEX and its tracks and describes details of the 2012 and 2013 Tweet Contextualization tracks. The general formats of the tweets and document collection and submission format for summaries are also described briefly.

### 2.1 INEX

The ***IN****itiative for the* ***E****valuation of* ***X****ML Retrieval* (INEX) [1] is a global forum facilitating research in the field of hypertext Information Retrieval (IR) since 2002. It provides the basic infrastructure required for the development and evaluation of research in IR. Traditionally, IR systems provide users with documents which may be relevant to their queries, and the task of accessing the exact information required from those documents is left to the user [1]. INEX facilitates the development of hypertext IR systems, which extract the required information automatically in the form of elements. INEX conducts a global competition each year, wherein various research tasks are presented to participants in the form of separate tracks. In 2012, INEX organized five such tracks, i.e., Relevance Feedback, Tweet Contextualization, Snippet Retrieval, Linked Data and Social Book Search. INEX 2013 maintains all the tracks from 2012 except for Relevance Feedback. Over 100 universities and organizations have participated in these tracks over these two years [2]. We chose the 2012 and 2013 Tweet Contextualization tracks as the source of our hypertext IR experiments.

### 2.2 Tweet Contextualization Track at INEX

With the advent of blogging, a major revolution has occurred in the way people express themselves on the Internet. Microblogging is one form of blogging, very famous in recent years, wherein people write short messages to convey their thoughts or opinions. *Twitter* is a leading microblogging website, which combines certain aspects of microblogging and social networking. Users of *Twitter* can post their short messages, called as tweets,

on the Internet. These tweets have a maximum length of 140 characters and can contain links to other forms of multimedia such as videos or images.

The tweets, with their 140-character length restriction, are rarely self-contained, i.e., they carry less information within themselves and it is difficult to guess the context of a tweet just by looking at its contents. The Tweet Contextualization track aims to provide users with additional information about these tweets. This additional information is to be presented to the user in the form of a summary [5]. It must be automatically extracted from a specified Wikipedia XML document collection. The Tweet Contextualization track was initiated by INEX in 2010 as a Question Answering task and gradually transformed into the Tweet Contextualization task.

INEX provides users with a Wikipedia-based XML document corpus and a set of queries in the form of tweets. It evaluates the summaries submitted by participants using uniform evaluation measures as described in Section 2.6.

## 2.3 Document Collection

The document collections for the Tweet Contextualization track are extracted from official dumps of English Wiki pages released by Wikipedia itself. INEX builds an XML document for each Wiki page present in the official dump. The document collection for the 2012 Tweet Contextualization track is based on the November 2011 official dump, whereas the document collection for the 2013 Tweet Contextualization track was extracted from the November 2012 official dump. The XML documents built by INEX are structured documents; i.e., they follow a specific Document Type Definition (DTD) [5]. Figure 1 shows the DTD for a typical Wiki XML page.

Each XML page has a unique identifier in the form of ID tag (*ID)*, a title tag (*title*), an abstract tag (*a*), and multiple section(*s*) and sub-header (*h*) tags. Each section can have multiple paragraph (*p*) tags, which contain actual text for a Wiki page. All notes and bibliographical references are removed from the Wikipedia dumps before generating the

XML representation [5]. Figure 2 shows a sample XML Wiki page in the 2012 document collection.

```
<!ELEMENT xml (page)+>

<!ELEMENT page (ID, title, a, s*)>

<!ELEMENT ID (#PCDATA)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT a (p+)>

<!ELEMENT s (h, p+)>
```

**Figure 1: DTD for a Wiki XML Page**

```
<xml>
<page>
<ID>1000</ID>
<title>Antlia</title>
<a>
<p o="1">
Antlia was created in 1756 by the French astronomer Abbé
<t>Nicolas Louis de Lacaille</t>, who created fourteen constellations
...
</p>
</a>
<s o="1">
<h>Deep-sky objects</h>
<p o="1">
Because it occupies a part of the <t>celestial sphere</t> that
faces away from the <t>Milky Way</t>,
Antlia contains very few deep-sky objects.
It contains ...
</p>
<p o="3">The <t>Antlia Dwarf</t>,
a 14.8m <t>dwarf spheroidal galaxy</t>
...
</p>
</s>
</page>
</xml>
```

**Figure 2: A Sample XML Wiki Page (1000.xml) from the 2012 Document Collection.**

## 2.4 Queries

The queries for the Tweet Contextualization track are provided in the form of tweets. INEX collects a set of public tweets posted by *Twitter* users. These tweets are from informative user accounts, e.g., news channels like *CNN* or magazines like *People*. Tweets from informative accounts avoid personal tweets, for which contextualization cannot be done [5].

Each tweet is identified by a unique tweet id. These tweets may consist of URLs, user name tags, and special characters like emoticons specifying users' sentiments and hashtags. Hashtags serve as the subject or headline of the tweets and typically start with the '#' symbol. Most hashtags are composed of more than one term, which are not separated by delimiters such as space (for example, *#USPresidentialElection2012*).

All tweets are provided in plain text format along with their ids. Figure 3 shows some sample tweets from the 2013 query set.

303481535074549763 *"007 in #SKYFALL's Floating Dragon casino. #SKYFALL IS OUT ON BLU- RA ……"*

306715982796292096 *"17 Afghan Police Officers Drugged and Killed http://t.co/i6uHFEUUlS"*

306252681373175808 *"Anne, Jennifer and Adele look on as their #Oscars2013 statues are engraved……."*

**Figure 3: Sample Query Tweets from the 2013 Query Set**

## 2.5 Submission Format

All participants of the Tweet Contextualization track must follow a specific format while submitting their summaries to INEX. The format of submissions for a summary is shown in Figure 4. A summary generated for a tweet is composed of multiple passages.

<tid> Q0 <file> <rank> <rsv> <run_id> <text of passage 1>

<tid> Q0 <file> <rank> <rsv> <run_id> <text of passage 2>

**Figure 4: Submission Format for the 2012 and 2013 Tweet Contextualization Track**

Table 1 describes the fields shown in Figure 4.

| Field | Description |
|---|---|
| <tid> | *The ID of the tweet, for which the summary is generated* |
| Q0 | *A field used by organizers for evaluation purpose. It has a fixed value of Q0* |
| <file> | *The ID of the document, from which the current passage is extracted* |
| <rank> | *The rank of the current passage in the summary* |
| <rsv> | *The calculated score for the current passage in the summary* |
| <run_id> | *A unique identifier for the participant's run* |
| <text of passage n> | *Actual text of passage for a summary* |

**Table 1: Summary of Fields in the Tweet Contextualization Track Submission Format**

An example of a summary following the above submission format is shown in Figure 5.

306715982796292096 Q0 8912201 1 0.3000 UMD65_sbarun *The ANP will continue growing ….*

306715982796292096 Q0 4872967 2 0.2632 UMD65_sbarun *The Afghan police force was first ….*

306715982796292096 Q0 8912201 3 0.2500 UMD65_sbarun *Between 2002 and 2009, the …*

**Figure 5: A Sample Summary in the Tweet Contextualization Track Submission Format**

## 2.6 Evaluation

All summaries submitted by participants for the 2012 and 2013 Tweet Contextualization tracks are evaluated based upon two metrics, namely, informativeness of summaries and readability of summaries.

### 2.6.1 Informativeness

The process of extracting summaries for Tweet Contextualization involves finding the highest related documents for a tweet, extracting the most informative passages from those documents and building of an automatic summary using those passages. Thus, the final summary generated must reflect not only the documents but also give the most comprehensive answer generated by QA and automatic summarization techniques. Thus informativeness of the summaries cannot be evaluated by using standard IR measures like Recall and Precision [13]. Instead, the Tweet Contextualization track uses Kullback-Leiber (KL) [7] and Jenssen-Shanon (JS) (Information Radius) [9] divergences for evaluating the informativeness of summaries submitted by participants. Informativeness is automatically evaluated by a toolkit provided by organizers. This toolkit uses a set of reference summaries created by the organizers for all tweets. The toolkit is based on a Porter stemmer and implements a new, normalized ad-hoc dissimilarity formula [13] as shown in Figure 6.

Organizers used 3 different distributions for the reference summaries in the 2012 and 2013 tracks. These distribution types are as follows [13]:

1. Unigrams made of single lemmas (after removing stop-words).
2. Bigrams made of pairs of consecutive lemmas (in the same sentence).
3. Bigrams with 2-gaps also made of pairs of consecutive lemmas but allowing the insertion between them of a maximum of two lemmas (Also referred to as skip distribution).

Thus, the informativeness metric measures the overlap of participants' summaries with the reference summaries by considering the number of overlapping passages, unigrams and bigrams present or missing in the summaries.

$$Dis(T,S) = \sum_{t \in T} \frac{f_T(t)}{f_T} \times \left(1 - \frac{\min(\log(P), \log(Q))}{\max(\log(P), \log(Q))}\right)$$

Where,

$$P = \frac{f_T(t)}{f_T} + 1$$

$$Q = \frac{f_S(t)}{f_S} + 1$$

$T -$ Set of query terms present in reference summary and

for each $t \in T$, $f_T(t)$ is the frequency of term t in reference summary

$S -$ Set of query terms present in a submitted summary and

for each $t \in S$, $f_S(t)$ is the frequency of term t in a submitted summary

**Figure 6: Dissimilarity Formula used in Informativeness Evaluation of Summaries**

### 2.6.2 Readability

INEX has followed different methodologies for the readability evaluations of participants' summaries for the 2012 and 2013 Tweet Contextualization tracks.

Readability evaluation for the 2012 Tweet Contextualization was a manual evaluation of participants' summaries, wherein a summary submitted by a participant was presented to other participants anonymously through an online web interface. Each summary is composed of multiple passages, and the assessor evaluates the readability of passages by considering following four parameters [13]:

1. Syntax: This parameter is used to check if a passage has segmentation problems (for example, if the sentence boundaries are not properly demarcated).
2. Anaphora: This parameter is used to check if there exist any unresolved co-references.
3. Redundancy: This parameter verifies redundant passages in a summary.
4. Trash: Marking a passage as trash indicates that it is totally irrelevant to the query. If a passage is marked as trash, then the other three parameters are not considered for evaluation of that passage.

For the 2013 Tweet Contextualization track, readability evaluation was performed by the organizers themselves. No feedback from other participants was considered for evaluating readability. Organizers finalized ten tweets from the 2013 query sets with the largest text references (t-rels), based on informativeness evaluation done earlier [13]. The parameters used by organizers for readability evaluation, namely, syntax, anaphora, redundancy and trash, remain the same from the 2012 track.

# 3. Implementation

Our QA system makes use of multiple IR and Natural Language Processing (NLP) tools such as Indri, Apache Lucene, Stanford CoreNLP and MontyLingua. This chapter gives a brief overview of these tools and their uses in our QA system. We then provide a detailed architecture of our QA system and a description of all steps followed in generation of automatic summaries for the Tweet Contextualization task.

## 3.1 Indri Text Search Engine

Indri is an open source search engine, which is a part of Lemur project maintained by Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts, Amherst, and the Language Technologies Institute (LTI) at Carnegie Mellon University [16]. Indri provides a powerful indexer capable of indexing a variety of document formats such as HTML, XML, PDF, plain text and TREC documents and an easy-to-use retrieval framework, which can be used for field and passage retrieval [16]. We use Indri as the primary indexing tool for our QA system.

The retrieval model in Indri combines language modeling and inference network approaches for the purpose of information retrieval [15]. The inference network approach assumes that a user query is a series of concepts. A concept can be a single term, a phrase or a more complex entity [15]. A document is considered relevant to the user query if it contains the concepts present in the user query, for example, if a user query consists of a phrase concept *Achilles' heel*, then the documents containing this concept are considered related to that query by Indri.

As mentioned above, Indri can be used to perform field retrieval from structured documents, meaning that we can restrict Indri to retrieve only those documents in which concepts from user queries appear in a particular field. For example, if we are interested in searching for the phrase *Vincent van Gogh*'*s paintings* only within the *body* element of an XML document, then we can restrict Indri retrieval only to the field *body*. This is a

useful feature for our QA system since all meaningful information is contained within *p* tags of the Wikipedia corpus provided by INEX.

## 3.2 Apache Lucene

Apache Lucene is another open source Java search library, which is written and maintained by Apache Software Foundation (ASF) [3]. It offers a powerful Information Retrieval framework which can accomplish a variety of tasks such as indexing, retrieval, spell checking, highlighting and tokenization. We use Lucene for the purpose of Hashtag Separation, described in detail in Section 3.5.4.

The default retrieval framework in Lucene is based on Vector Space Model (VSM) and it can be extended to other models such as BM25 or BM25F probabilistic models [3]. Lucene produces an inverted index (a mapping of terms to the documents in which those terms appear) [3]. We use the default VSM in our experiments. The VSM [12] treats documents and queries as vectors. First, a stop-list for non-substantive terms is used to delete common words from documents. Then using techniques such as automatic suffix stripping, word stems are produced as vectors. The *tf-idf* weights [12] are calculated for all vectors. Thus, each document is represented as term vector along with corresponding term weights. Similarity between document and query vectors is calculated by using a similarity measure, such as *cosine*.

The Lucene index can also be queried to yield the frequency of the query terms or phrases. For example, we can query the Lucene index to get the total number of times a phrase such as *Gandalf the grey* appears. We use this feature of Lucene in the Hashtag Separation component of our QA system.

## 3.3 Stanford CoreNLP

Stanford CoreNLP [17] is natural language analysis tool for the English language, comprised of features such as lemmatization, sentence boundary detection, Named Entity

Recognition (NER) and part-of-speech (POS) tagging. It is developed by the NLP group at Stanford University.

We use the lemmatization and sentence boundary detection features of the Stanford CoreNLP tool in our QA system. Lemmatization is a process reducing a word into its lemma, which is the dictionary or base form of that word, for example, words like *runs* and *running* will be reduced to their dictionary form *run*. It is used to reduce the inflectional forms of a word [8]. This normalization facilitates information retrieval. We lemmatize the individual terms from the tweets before using them for retrieval purposes in our QA system.

Sentence boundary detection recognizes the demarcation at the end of a sentence. Sentence boundary detection is challenging because punctuation at the end of sentences can be ambiguous. A sentence may end in question mark (?) or exclamation mark (!). A period (.) may not always signify the end of sentence since it can be a part of an abbreviation (e.g., *Dr.*) or a decimal number (e.g., *1.256*). The 500-word context summary required by the Tweet Contextualization track is composed of multiple sentences extracted from the Wikipedia corpus. A mechanism to extract sentences with proper boundaries is required in order to produce readable sentences from Wiki pages. A robust mechanism is achieved by using the sentence boundary detection feature of the Stanford CoreNLP.

### 3.4 MontyLingua

MontyLingua [10] is a natural language understander tool for English, developed by MIT Media Labs at Massachusetts Institute of Technology. It consists of a tokenizer, a part-of-speech (POS) tagger and a chunker (noun phrase detector) for the English language text. Raw text can be fed as an input to MontyLingua and it can generate a semantic representation of the text in terms of tokens, POS tags and noun phrases.

We use MontyLingua in our QA system to extract noun phrases from the tweets. Noun phrases can be named entities such as names of persons, locations, organizations,

quantities, etc. These noun phrases serve as keywords from the tweets and are useful in extracting exact matching documents from the Wikipedia corpus.

## 3.5 Architecture of QA system for Tweet Contextualization Task

Our QA system, leveraged with IR and NLP tools, performs indexing, retrieval and summary generation operations for the Tweet Contextualization task in multiple steps. Figure 7 shows the architecture of the QA system with a workflow of steps resulting in automatic context summary generation.

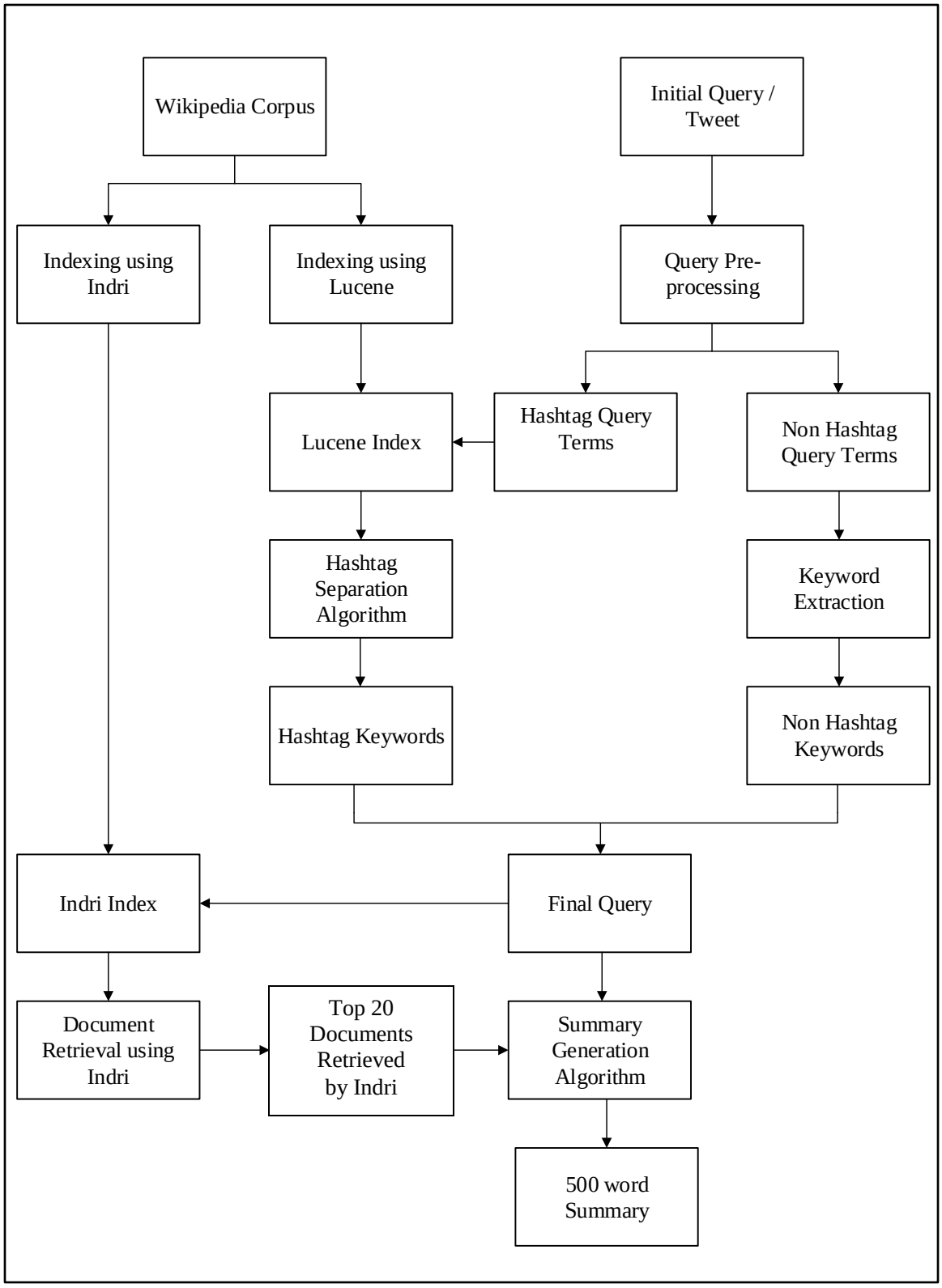


**Figure 7: Architecture of Question-Answering System for Tweet Contextualization**

The following sections provide the details of all stages shown in Figure 7.

### 3.5.1 Indexing Wiki using Indri

As the first step of context summary generation, we index the Wikipedia corpus using Indri. Since Indri is capable of indexing XML documents directly and the Wiki pages in the 2011 and 2012 document collections are structured XML documents, no preprocessing of documents is required. Indri takes two files as input parameters, namely, a configuration file specifying the path of corpus to be indexed, fields to be indexed, memory requirements, stemmer (Porter or Krovetz), and another file consisting of a list of stop-words. We choose only three fields to index - *title* (title tag), *h* (header tag) and *p* (paragraph tag) since other tags do not contain any information useful for retrieval. We use the Porter stemmer to convert document terms into word stems. We choose a list of common English words such as *the*, *about, being* as a stop-word list. These stop-words are ignored by Indri while indexing. Indri creates an index of Wiki pages in binary file format. This index is used as the primary index for document retrieval.

### 3.5.2 Indexing Wiki using Lucene

After the creation of the Indri index, the same Wikipedia corpus is passed to Lucene for creation of a separate Lucene index. Lucene also takes the path of the corpus to be indexed and the stop-word list as input. It creates an inverted index of Wiki pages specifying the mapping of terms to the documents which contain them.

The Lucene index is created as an output of this step and is used as a secondary index by our QA system for Hashtag separation purposes.

### 3.5.3 Query Preprocessing

As a next step, we preprocess the queries or tweets. Each individual tweet is fetched from (the 2012 and 2013) query sets and passed to the preprocessing step. We remove URLs from each tweet as URLs are insignificant for the purpose of contextualization.

Information fetching from the URLs present in tweets is prohibited by INEX organizers [5], so URLs are not to be further used in automatic summary generation.

Tweets generally have user and Retweet tags. User tags (which start with @ symbol) and Retweet tags (which consist of the *RT* characters) signify the names of twitter users and are not specifically related to the tweet. Hence, we remove these two tags from the tweets. We also discard certain special characters from tweets such as emoticons and XML character entities since these characters do not carry any useful information. Next, we separate out hashtags terms (which start with the # symbol) from the tweets. Hashtags are important terms, which specify the topic of the tweet and are useful in understanding its meaning. We remove the preceding # symbol from the Hashtag. The remaining non-hashtag terms are further processed to remove stop words. Finally, we lemmatize the non-hashtag terms using the Stanford CoreNLP to produce dictionary forms of those terms. Thus the query preprocessing step of our QA system generates preprocessed Hashtag and non-hashtag terms.

Figure 8 shows two sample tweets from the 2013 query set before the query preprocessing stage and Figure 9 shows the Hashtag and non-hashtag terms generated for those tweets after the preprocessing stage.

| | |
|---|---|
| *304283818020450304* | *RT @American_Heart: Warning signs are different in men & women. Ask @American_Heart @BaylorHealth @DailyRX #HeartChat, Feb. 21, 12:30CST http://t.co/b4iQLz55* |
| *306099230823567362* | *RT @danellecheney: view from my desk at @cincyartmuseum, for the @Tate :) downtown #cincinnati and a view of @CincyPlay! http://t.co/N65mG6c8kw* |

**Figure 8: Sample Tweets from 2013 Query Set before Query Preprocessing**

***For Tweet 304283818020450304:***

**Hashtag Terms:** *HeartChat*

**Non-hashtag Terms:** *warning sign different man woman ask Feb 21 12:30 cst*

***For Tweet 306099230823567362:***

**Hashtag Terms:** *cincinnati*

**Non-hashtag Terms:** *view desk downtown view*

**Figure 9: Hashtag and Non-hashtag Terms Generated after Query Preprocessing**

### 3.5.4 Hashtag Separation Algorithm

As previously mentioned, Hashtags are important terms as they signify the topic of a tweet and so play a pivotal role in the tweet contextualization. The tweets from the 2013 query set are rich with Hashtags. A single tweet can contain multiple instances of Hashtags. *Twitter* Hashtags are generally composite terms wherein individual terms are written consecutively without delimiters. For example, the Hashtag *#Recipeoftheday* consists of four terms (*Recipe, of, the, day*), which are present without any delimiters. This calls for a mechanism for word segmentation within the Hashtag in order to achieve Hashtag composite term separation.

A naïve approach to the problem of Hashtag separation might try to separate the composite terms by using a predefined closed vocabulary. One could try to find the maximum overlap between a dictionary word and a Hashtag term and separate out individual terms. This approach suffers two drawbacks. First, it is not possible to build a closed vocabulary of all words which can form *Twitter* Hashtags. Given the ever evolving nature of the English language, new words are added on daily basis and new Hashtags can emerge. So, building a closed vocabulary for this purpose is virtually impossible. Second, a vocabulary-based approach may not guarantee correct word segmentation within a Hashtag. For example, a Hashtag *#choosespain* can be interpreted as a composite of *choose*s and *pain* or of *choose* and *spain*, since both of these combinations correspond to valid vocabulary terms.

We deal with the problem of Hashtag separation using an elegant approach to word segmentation suggested by Norvig [11]. We modify this approach as follows:

1. First, we start creating possible candidate segmentations for a Hashtag. We observe that the maximum number of terms in a Hashtag for the INEX query set is 4. We create candidate segmentations by inserting 0-4 spaces at all possible positions in a Hashtag string. For example, Table 2 shows some of the possible candidates created for Hashtag *#USPresidentialElection2012* with 0, 1, 2, 3 and 4 spaces inserted in it.

| Space Count | Possible Candidate Segmentations |
|---|---|
| 0 | *USPresidentialElections2012* |
| 1 | *U SPresidentialElections2012, US PresidentialElections2012, USP residentialElections2012, USP residentialElections2012, USPr esidentialElections2012 …..* |
| 2 | *U S PresidentialElections2012, U SP residentialElections2012, U SPr esidentialElections2012, U SPr esidentialElections2012, U SPr esidentialElections2012 …..* |
| 3 | *U S P residentialElections2012, U SP r esidentialElections2012, U SPr e sidentialElections2012, U SPr e sidentialElections2012, U SPr es identialElections2012 …..* |
| 4 | *U S P r esidentialElections2012, U SP r e sidentialElections2012, U SPr e s identialElections2012, U SPr e s identialElections2012, U SPr es i dentialElections2012 …..* ***US Presidential Elections 2012*** *….. USPresidentialElection s 2 0 1 2* |

**Table 2: Some of the Possible Candidate Segmentations generated by Hashtag Separation Algorithm for Hashtag *#USPresidentialElection2012***

2. To determine the best candidate segmentation out of all possible candidates, a probabilistic language model is used. This model predicts the best candidate segmentation by checking frequencies of each candidate in our Wikipedia corpus. We use the Lucene index in order to get frequencies of all candidates and chose the one with highest frequency as the best candidate segmentation.

   This probabilistic language model can be represented mathematically as shown in Figure 10. The best candidate chosen for Hashtag *#USPresidentialElection2012* using this approach is highlighted in Table 2.

$$\overline{c} = argmax_{c \in C}\, Count(c)$$

*where,*

$C =$ *Set of all possible candidate segmentations for a Hashtag,*

$\overline{c} =$ *Best candidate segmentation,*

$Count(c)$ $=$ *Frequency of a candidate segmentation c in Wiki Corpus*

*and* $argmax_x\, f(x)$ *means "the x such that* $f(x)$ *is maximized."*

**Figure 10: Mathematical Representation of Language Model used in Hashtag Separation Algorithm**

This approach for Hashtag separation ensures the correct segmentation for ambiguous hashtags such as *#insufficientnumbers*, which can be interpreted as *insufficient numbers* or *in sufficient numbers*. The candidate *in sufficient number* will be chosen as best segmentation by our language model since it has higher frequency in Wikipedia corpus. Each term from the best candidate is regarded as a keyword from the Hashtag. Thus, the Hashtag Separation stage of our QA system generates Hashtag keywords as output.

### 3.5.5 Keyword Generation for Non-hashtag Query Terms

Next, we extract keywords from the non-hashtag terms generated by the query preprocessing stage. We use MontyLingua as a tool for this purpose. The non-hashtag terms are given as an input to MontyLingua, and it generates noun phrases such as the names of people, locations, dates, etc., as the keywords. Figure 11 shows a sample of the keywords generated from non-hashtag terms for a tweet from the 2013 query set.

**Original Tweet**

*Also on Saturday at Istanbul's Istinye Park Shopping Mall will be a Guinness Book of Records attempt by two football freestylers. (2/2)*

**Non-hashtag Terms Generated by Query Preprocessing Stage**

*Saturday Istanbul Istinye Park Shopping mall Guinness Book Records attempt two football freestyler 2/2*

**Keywords Generated by Non-hashtag Terms**

*'Saturday', 'Istinye Park Shopping mall', 'Guinness Book Records',*

*'football freestyler', '2/2'*

**Figure 11: Keywords Generated for Non-hashtag Terms from a Sample 2013 Query**

### 3.5.6 Final Query Generation

Next, we combine the keywords generated from Hashtag terms and non-hashtag terms to form the final query. This query will be used to retrieve documents from Wikipedia corpus.

### 3.5.7 Document Retrieval using Indri

Using the final query generated in the previous stage, we query the Indri index created in the earlier stage of our QA system. The final query is used to retrieve the top 20 documents in a rank order. We restrict the retrieval process to the document field *p* (paragraph).

### 3.5.8 Summary Generation Algorithm

Next, we extract the 500-word context summary from the top 20 documents returned by Indri. This context summary is made up of sentences present in Wiki pages. Here, we exercise sentence retrieval facilitated by the sentence boundary detection feature of the Stanford CoreNLP tool. We rank the sentences retrieved from Wiki pages using a simple

arithmetic scoring technique, which considers the overlap of n-grams from the final query with n-grams in individual sentences.

Figure 12 shows an example of a plain text format context summary generated for a tweet from the 2013 query set. The actual summary submitted to INEX follows the submission format [5] required for the Tweet Contextualization track.

**Query / Tweet**

*24 hours to end the bankers tax dodge -- Tell EU ministers to make them pay their fair share. Sign now & RT: http://t.co/3AJrL3hd4J*

***The* context summary generated for above query**

*The emergency measures had been deemed unnecessary by the Icelandic government less than 24 hours earlier. In regards of Ireland, Portugal and Greece they on the other hand have not yet regained complete market access, and thus do not yet qualify for OMT support. Dietlein, Georg (2012): National Approaches towards a Financial Transaction Tax and Their Compatibility with European Law, EC Tax Review, Vol. As such, this tax is neither a financial activities tax, nor a "bank tax", for example. Companies cannot claim expenses in high tax jurisdiction, while profits are taxed in low tax jurisdictions. The scheme makes it impossible for say French or German banks to avoid the tax by moving their transactions offshore, unless they give up all their European customers. The most frequently discussed versions of a currency transaction tax are the Tobin tax and Spahn tax. The European Union financial transaction tax (EU FTT) is a proposal made by the European Commission to introduce a financial transaction tax (FTT) within the 27 member states of the European Union by 2014. The Automated Payment Transaction tax (APT tax) taxes the broadest possible tax base, namely all transactions including all real and financial asset transactions. To avoid the withholding tax, certain types of individuals could also prove that they were exempt from taxation in their country of residence. The following day the European Commission called for Tobin-style taxes on the EU's financial sector to generate direct revenue for the European Union. The group pays tax on the share of profit apportioned to the Slovak Republic at the Slovak tax rate (19%), while tax on the German share of profit is paid at the German tax rate (30%). On the other hand, tax administrations have to operate an additional tax system if the current tax code is maintained (parallel tax base, CCCTB). It might be the case that some tax administrations find their workload increased by the new tax rules. The retention tax agreed with the European Union (EU) in the taxation of savings income agreement is a suitable and efficient means of doing so. As a result, various new forms of financial transaction taxes were proposed, such as the EU financial transaction tax. Finally, every allocated share of profit is taxed in the respective Member State with the relevant corporate tax rate C .. The initial APT tax proposal was envisioned as a revenue neutral replacement for the entire Federal tax system.EU member states may decide to increase their part of the revenues by taxing financial transactions at a higher rate. According to some economists, a financial transaction tax is less susceptible to tax avoidance and tax evasion than other types of taxes proposed for the financial sector.*

**Figure 12: A Sample 500 Word Context Summary Generated for Tweet Id 306706888576360449 from 2013 Query Set**

# 4. Results

This chapter describes the process of context summary generation from the top 20 documents retrieved from the Wikipedia corpus using Indri. We then describe the details of the informative and readability evaluations and statistics of reference summaries used by INEX in the evaluation process. We also present results achieved by our QA system after the INEX evaluation of the 2012 and 2013 Tweet Contextualization tracks.

## 4.1 Context Summary Generation

A context summary as defined by INEX is " a readable text that provides some context about the subject of the tweet, in order to help the reader to understand it, i.e., answering questions of the form *What is this tweet about?*" [5]. As mentioned in Chapter 2, this context summary is 500 words in length and is composed of multiple passages extracted from the Wikipedia corpus. We extract the basic passages for building context summaries in the form of sentences. We view each passage as a sentence and multiple sentences are extracted from the top 20 documents (as retrieved by Indri) to form the final 500 word summary. The context summary generation process is detailed below.

> **Step 1:** The final query, generated from the Hashtag keywords and non-hashtag keywords from a tweet, is divided into individual terms. Each term is regarded as a token for summary generation purposes.
>
> E.g., if the final query is *Guinness Book Records,* then the tokens produced from it are *Guinness, Book* and *Records.*
>
> **Step 2:** Generate all possible combinations of tokens fetched in Step 1. These different combinations produce n-grams for the query.
>
> E.g., Table 3 shows the unigrams, bigrams and trigrams produced from tokens *Guinness, Book* and *Records*.

| | |
|---|---|
| **unigram** | *Guinness*<br>*Book*<br>*Records* |
| **bigrams** | *Guinness Book*<br>*Guinness Records*<br>*Book Guinness*<br>*Book Records*<br>*Records Guinness*<br>*Records Book* |
| **trigrams** | *Guinness Book Records*<br>*Guinness Records Book*<br>*Book Guinness Records*<br>*Book Records Guinness*<br>*Records Guinness Book*<br>*Records Book Guinness* |

**Table 3: N-grams Produced in Context Summary Generation Process for a Sample Query *Guinness Book Records***

**Step 3:** Retrieve all sentences from *p* tag (paragraph) of top 20 documents.

**Step 4:** For each sentence retrieved in Step 3

a. If the length of the sentence is less than a threshold value (N), then go to Step 3.

b. Initialize the sentence score to 0.

c. For each N-gram generated from query, check if it is present in the sentence.

i. If yes, then increment the sentence score by the length of N-gram.
ii. Else, go to Step 4.c.

d. Normalize the sentence score by dividing it by total length of the sentence.

**Step 5:** Sort all sentences in descending order of their scores.

**Step 6:** Extract the top-ranked sentences falling within a 500 word buffer to form the context summary.

We consider only the sentences with length greater than the threshold value while forming context summaries in order to avoid shorter sentences, which may not be informative.

## 4.2 Tweet Contextualization Experiments

Our experiments for the 2012 and 2013 Tweet Contextualization tracks were centered on the hashtag separation strategy and the summary generation process.

Hashtags were underused in the 2012 query set while the tweets in the 2013 query set were rich in hashtags. Organizers increased the diversity of tweets for the 2013 track, and a significant part of that query set contained hashtags. Since the hashtag was not an important factor in 2012, no specific processing for hashtag separation was done that year. We introduced the hashtag separation component in 2013 for the Tweet Contextualization experiments.

We experimented with different values for the threshold length of sentences considered in the context summary generation process. For the 2011 Wikipedia corpus, we observed that threshold values of 10 and 20 produced better results. INEX allows each participant organization to submit a maximum of three runs each year for the Tweet Contextualization track. We submitted two runs, using threshold values of 10 and 20, in 2012. For the 2012 Wikipedia corpus (provided by INEX for the 2013 track), we finalized the threshold value for sentence length in the generated summary at 15. We submitted one run with a threshold value of 15 for the 2013 track.

## 4.3 Results of INEX Evaluations for the Tweet Contextualization Tracks

As seen in Chapter 2, INEX assesses context summaries based on informativeness and readability. This section provides details of the 2012 and 2013 Tweet Contextualization evaluations and results achieved by our system for those evaluations.

### 4.3.1 Informativeness for 2012

For 2012, organizers built a reference summary set for 63 tweets from the 2012 query set. The tweets which contain adequate contextual information in the 2011 Wikipedia corpus were considered for building the reference run. Table 4 shows the statistics of the 2012 reference set used in informativeness evaluation.

| No. of tweets in reference set | Total passages | Total number of tokens |
|---|---|---|
| 63 | 3471 | 104151 |

**Table 4: Statistics of 2012 Reference Summary Set for the Informativeness Evaluation**

Using the dissimilarity formula specified in Figure 6 and the unigram, bigram and skip distribution types described in section 2.6.1, INEX evaluated 33 runs from all participants. Two result runs generated by our QA systems with IDs 178 and 152 ranked first and second, respectively, in the informativeness evaluation. Table 5 shows the unigram, bigram and skip distribution scores achieved by the top 10 runs for 2012 informativeness evaluation ranked according to the skip distribution score.

| Rank | Run ID | Unigram Distribution Score | Bigram Distribution Score | Skip Distribution Score |
|---|---|---|---|---|
| **1** | **178** | **0.7734** | **0.8616** | **0.8623** |
| **2** | **152** | **0.7827** | **0.8713** | **0.8748** |
| **3** | 170 | 0.7901 | 0.8825 | 0.8848 |
| **4** | 194 | 0.7864 | 0.8868 | 0.8887 |
| **5** | 169 | 0.7959 | 0.8881 | 0.8904 |
| **6** | 168 | 0.7972 | 0.8917 | 0.8930 |
| **7** | 193 | 0.7909 | 0.8920 | 0.8938 |
| **8** | 185 | 0.8265 | 0.9129 | 0.9135 |
| **9** | 171 | 0.8380 | 0.9168 | 0.9187 |
| **10** | 186 | 0.8347 | 0.9210 | 0.9208 |

**Table 5: Top 10 Runs in the 2012 Tweet Contextualization Informativeness Evaluation**

### 4.3.2 Readability for 2012

Readability evaluation for the 2012 Tweet Contextualization track was a manual evaluation performed by participants. Organizers selected the same pool of 63 tweets from the informativeness evaluation, and the summaries submitted by participants for these 63 tweets were presented for evaluation to other participants through a web-based interface. Participants were instructed to assess each passage in the summary considering the four basic parameters syntax, anaphora, redundancy and trash (see Section 2.6.2). Our two runs, with IDs 152 and 178, ranked 8 and 10, respectively, on the official readability evaluation. Table 6 shows the score achieved by the top 10 runs along with our runs, ranked 8 and 10.

| Rank | Run | Score for Trash Parameter | Score for Syntax Parameter | Combined score for Anaphora and Redundancy Parameter |
|---|---|---|---|---|
| **1** | 185 | 0.7728 | 0.7452 | 0.6446 |
| **2** | 171 | 0.631 | 0.606 | 0.6076 |
| **3** | 168 | 0.6927 | 0.6723 | 0.5721 |
| **4** | 194 | 0.6975 | 0.6342 | 0.5703 |
| **5** | 186 | 0.7008 | 0.6676 | 0.5636 |
| **6** | 170 | 0.676 | 0.6529 | 0.5611 |
| **7** | 165 | 0.5936 | 0.6049 | 0.5442 |
| **8** | **152** | **0.5966** | **0.5793** | **0.5433** |
| **9** | 155 | 0.6968 | 0.6161 | 0.5315 |
| **10** | **178** | **0.6336** | **0.6087** | **0.5289** |

**Table 6: Results of the 2012 Tweet Contextualization Readability Evaluation**

### 4.3.3 Informativeness for 2013

For the 2013 Tweet Contextualization track, organizers finalized three reference sets for informativeness evaluation, namely, (1) Prior - a set of relevant pages selected by organizers while building the 2013 topics, (2) Pool - a set of selection of most relevant passages from participant submissions for tweets selected by organizers, and (3) All – a set of relevant texts merged with an extra selection of relevant passages from a random

pool of ten tweets [5]. Table 7 shows the statistics of the 2012 reference set used for informativeness evaluation [5].

| Reference set type | No. of tweets in reference set | Total passages | Total number of tokens |
|---|---|---|---|
| **Prior** | 40 | 380 | 11523 |
| **Pool** | 45 | 1760 | 58035 |
| **All** | 70 | 2378 | 77043 |

**Table 7: Statistics of 2013 Reference Summary Sets for the Informativeness Evaluation**

Participants' summaries were evaluated against all three reference sets considering distribution types, namely, unigram, bigram and skip (similar to 2012 informativeness evaluation).

Organizers released results for 24 runs submitted by participants ranked in descending order of Skip Distribution score for the All reference set. Our run, with ID 254, ranked 7 on the official informativeness evaluation. Table 8 shows the scores of the top 10 runs along with scores achieved by our run. Table 9 shows the names of participant organizations who submitted these top 10 runs.

| Rank | Run ID | Skip Dist. Score | Bigram Dist. Score | Unigram Dist. Score | Skip Dist. Score | Bigram Dist. Score | Unigram Dist. Score | Skip Dist. Score | Bigram Dist. Score | Unigram Dist. Score |
|---|---|---|---|---|---|---|---|---|---|---|
| | | All Reference Set | | | Pool Reference Set | | | Prior Reference Set | | |
| **1** | 256 | 0.8861 | 0.8810 | 0.7820 | 0.8752 | 0.8700 | 0.7813 | 0.9210 | 0.9134 | 0.7814 |
| **2** | 258 | 0.8943 | 0.8908 | 0.7939 | 0.8802 | 0.8766 | 0.7916 | 0.9288 | 0.9226 | 0.7985 |
| **3** | 275 | 0.8969 | 0.8924 | 0.8061 | 0.8789 | 0.8745 | 0.7941 | 0.9172 | 0.9106 | 0.7899 |
| **4** | 273 | 0.8973 | 0.8921 | 0.8004 | 0.8802 | 0.8750 | 0.7923 | 0.9235 | 0.9155 | 0.7862 |
| **5** | 274 | 0.8974 | 0.8922 | 0.8009 | 0.8805 | 0.8751 | 0.7932 | 0.9234 | 0.9154 | 0.7872 |
| **6** | 257 | 0.8998 | 0.8969 | 0.7987 | 0.8916 | 0.8895 | 0.8010 | 0.9341 | 0.9280 | 0.7992 |
| **7** | **254** | **0.9242** | **0.9229** | **0.8331** | **0.9162** | **0.9159** | **0.8363** | **0.9473** | **0.9430** | **0.8223** |
| **8** | 276 | 0.9301 | 0.9270 | 0.8169 | 0.9333 | 0.9302 | 0.8285 | 0.9718 | 0.9678 | 0.8286 |
| **9** | 270 | 0.9397 | 0.9365 | 0.8481 | 0.9274 | 0.9246 | 0.8418 | 0.9686 | 0.9642 | 0.8529 |
| **10** | 267 | 0.9468 | 0.9444 | 0.8838 | 0.9389 | 0.9362 | 0.8802 | 0.9625 | 0.9596 | 0.8830 |

**Table 8: Results of the 2013 Tweet Contextualization Informativeness Evaluation**

| Rank | Run ID | Participant ID | Name of the Participant Organization |
|---|---|---|---|
| **1** | 256 | 199 | Université de Nantes |
| **2** | 258 | 199 | Université de Nantes |
| **3** | 275 | 182 | IRIT, Perm State National Research University |
| **4** | 273 | 182 | IRIT, Perm State National Research University |
| **5** | 274 | 182 | IRIT, Perm State National Research University |
| **6** | 257 | 199 | Université de Nantes |
| **7** | **254** | **65** | **University of Minnesota Duluth** |
| **8** | 276 | 62 | LIA - University of Avignon |
| **9** | 270 | 46 | Jadavpur University |
| **10** | 267 | 46 | Jadavpur University |

**Table 9: Participant Organizations Corresponding to Top 10 Runs in the 2013 Tweet Contextualization Informativeness Evaluation**

### 4.3.4 Readability for 2013

In 2013, INEX organizers performed the role of evaluators instead of participants for assessing the readability of participants' summaries. Organizers finalized a pool of 10 tweets with the largest text reference set from the 2013 informativeness evaluation. (For these 10 tweets, context summaries from participants were expected to have almost 500 words since the reference set was much larger [5].) Organizers evaluated the summaries based on the scores for syntax, anaphora, redundancy and trash similar to the 2012 readability evaluation, and released a ranking of 22 runs from all participants. The runs were ranked according to mean average value of all 4 scores for syntax, anaphora, trash and redundancy. Our run, 254, ranked 6 on the official evaluation. Table 10 shows the top 10 runs from the 2013 readability evaluations. Table 11 shows the names of participant organizations who submitted these top 10 runs.

| Rank | Run | Mean Average | % Score for Trash Parameter | % Score for Redundancy Parameter | % Score for Anaphora Parameter | % Score for Syntax Parameter |
|---|---|---|---|---|---|---|
| **1** | 275 | 72.44 | 76.64 | 67.30 | 74.52 | 75.50 |
| **2** | 256 | 72.13 | 74.24 | 71.98 | 70.78 | 73.62 |
| **3** | 274 | 71.71 | 74.66 | 68.84 | 71.78 | 74.50 |
| **4** | 273 | 71.35 | 75.52 | 67.88 | 71.20 | 74.96 |
| **5** | 257 | 69.54 | 72.18 | 65.48 | 70.96 | 72.18 |
| **6** | **254** | **67.46** | **73.30** | **61.52** | **68.94** | **71.92** |
| **7** | 258 | 65.97 | 68.36 | 64.52 | 66.04 | 67.34 |
| **8** | 276 | 49.72 | 52.08 | 45.84 | 51.24 | 52.08 |
| **9** | 267 | 46.72 | 50.54 | 40.90 | 49.56 | 49.70 |
| **10** | 270 | 44.17 | 46.84 | 41.20 | 45.30 | 46.00 |

**Table 10: Results of the 2013 Tweet Contextualization Readability Evaluation**

| Rank | Run ID | Participant ID | Name of the Participant Organization |
|---|---|---|---|
| **1** | 275 | 182 | IRIT, Perm State National Research University |
| **2** | 256 | 199 | Université de Nantes |
| **3** | 274 | 182 | IRIT, Perm State National Research University |
| **4** | 273 | 182 | IRIT, Perm State National Research University |
| **5** | 257 | 199 | Université de Nantes |
| **6** | **254** | **65** | **University of Minnesota Duluth** |
| **7** | 258 | 199 | Université de Nantes |
| **8** | 276 | 62 | LIA - University of Avignon |
| **9** | 267 | 46 | Jadavpur University |
| **10** | 270 | 46 | Jadavpur University |

**Table 11: Participant Organizations Corresponding to Top 10 Runs in the 2013 Tweet Contextualization Readability Evaluation**

## 5. Conclusions and Future Work

The results of the 2012 and 2013 INEX evaluations suggest that the summarization approach used in our QA system performs well on informativeness but would benefit from increased readability of context summaries. The approach for generating context summaries (i.e., considering N-gram overlap between queries and sentences from Wiki pages) helps ensure that context summaries contain adequate correlating information with the tweets and avoid inclusion of non-similar information in them as much as possible. We identify the cause of lower rank of the 2013 informativeness evaluation (as compared to that in 2012) as less accurate hashtag separation. A hybrid approach using a dictionary and probabilistic language model may produce better result for hashtag separation.

It may be that readability of context summaries can be improved by considering contiguous sentences from single documents rather than merging sentences from multiple documents, as is done in our system. Redundancy in the context may be reduced by considering various sentence similarity measures [4]. Another approach to improve readability of context summaries might be devised by introducing a mechanism for co-reference resolution [6] in order to avoid unresolved anaphora in context summaries. Initial work has been done on this front, but no conclusive results have been achieved.

Another interesting aspect of this problem is query expansion. Given the short nature of tweets, the queries generated by our QA systems are even shorter. The information content of queries may be enriched by considering query expansion.

We conclude that efficient mechanisms for query formation and summarization are essential for generating good context summaries. We anticipate that the future work will aim at improving these mechanisms in order to generate more informative and readable context summaries for the Tweet Contextualization task.

## 6. References


[1] About INEX [Internet]. Amsterdam, Netherlands. INEX: c 2008-2013 [cited 2013 Jun 20]. Available from: https://inex.mmci.uni-saarland.de/about.html

[2] Bellot P, Chappell T, Doucet A *et al*. Report on INEX 2012. In: *ACM SIGIR Forum*; 2012, Dec.: ACM. 46 (2): p. 50-59.

[3] Białecki A, Muir R, Ingersoll G. Apache Lucene 4. In: *SIGIR 2012 Workshop on Open Source Information Retrieval*; 2012. p. 17-24.

[4] Higgins D, Burstein J. Sentence similarity measures for essay coherence. In: *Proceedings of the 7th International Workshop on Computational Semantics*, Tilburg, Netherlands; 2007.

[5] INEX 2013 Tweet Contextualization Track [Internet]. Amsterdam, Netherlands. INEX: c 2008-2013[cited 2013 Jun 20].
Available from: https://inex.mmci.uni-saarland.de/tracks/qa/

[6] Jurafsky D, Martin J.H. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. 2nd ed. Upper Saddle River, N.J.: Prentice Hall; 2008. 704 p.

[7] Kullback S, Leibler R.A. On information and sufficiency. In: *The Annals of Mathematical Statistics*; 1951.22(1): p. 79-86.

[8] Manning C.D., Raghavan P, Schütze H. Introduction to information retrieval. 1st ed. Cambridge: Cambridge University Press; 2009. 57 p.

[9] Manning C.D., Schütze H. Foundations of statistical natural language processing. Cambridge: MIT press; 1999. 329 p.

[10] MontyLingua- A Free, Commonsense-Enriched Natural Language Understander for English [Internet]. Boston, MA. MIT Media Labs: c 2002-2013 [cited 2013 May 16]. Available from: http://web.media.mit.edu/~hugo/montylingua

[11] Norvig P. Natural language corpus data. In: Segaran T, Hammerbacher J. *Beautiful data: The stories behind elegant data solutions*. 1st ed. Sebastopol, CA: O'Reilly Media; 2009. p. 219-242.

[12] Salton G, Wong A, Yang C.S. A vector space model for automatic indexing. In: Communications of the ACM; 1975. 18(11): p 613-620.

[13] Sanjuan E, Moriceau V, Tannier X *et.al.* Overview of the INEX 2012 Tweet Contextualization track. In: Geva S, Kamps J, Schenkel R, editors. *Focused access to content, structure and context*. *Proceedings of 11th International Workshop of the Initiative for the Evaluation of XML Retrieval*; 2012.

[14] Song F., Bruce C.W. A general language model for information retrieval. In: *Proceedings of the eighth international conference on Information and knowledge management*; 1999: ACM. p. 316-321.

[15] Strohman T, Metzler D, Turtle H, Bruce C.W. Indri: A language model-based search engine for complex queries. In: *Proceedings of the International Conference on Intelligent Analysis*; 2005; 2(6): p. 2-6.

[16] The lemur toolkit for language modeling and information retrieval [Internet]. Pittsburgh PA. The Lemur Project: c 2000-2013 [cited 2013 May 18]. Available from: http://www.lemurproject.org/indri.php

[17] Toutanova K, Klein D, Manning C.D., Singer Y. Feature-rich part-of-speech tagging with a cyclic dependency network. In: *Proceedings of the 2003 Conference of the North American. Chapter of the Association for Computational Linguistics on Human Language Technology*; 2003: ACL. 1: p. 173-180.