UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

LEARNING TO RANK SOCIAL KNOWLEDGE FOR QUESTION ANSWERING IN STREAMING PLATFORMS

TESIS PARA OPTAR AL GRADO DE DOCTOR EN CIENCIAS, MENCIÓN COMPUTACIÓN

JOSE MIGUEL HERRERA MALDONADO

PROFESOR GUÍA:
BARBARA POBLETE LARA

PROFESOR CO-GUÍA:
DENIS PARRA SANTANDER

MIEMBROS DE LA COMISIÓN:
AIDAN HOGAN
MARCELO MENDOZA ROCHA
MOUNIA LALMAS

Este trabajo ha sido parcialmente financiado por CONICYT,
INSTITUTO MILENIO FUNDAMENTOS DE LOS DATOS.

SANTIAGO DE CHILE
2019

Resumen

Las plataformas de redes sociales han cambiado la forma de buscar y encontrar información en la web. En particular, los sitios de Preguntas y Respuestas (QA: Question Answering) han surgido como plataformas diseñadas específicamente para el intercambio de preguntas y respuestas entre las comunidades de usuarios. Si bien los usuarios tienden a encontrar respuestas de buena calidad en estos sitios, hay evidencia de que existe un volumen significativo de interacciones QA en plataformas sociales como Twitter. La literatura al día de hoy indica que los usuarios eligen este tipo de plataformas, no especializadas para QA, dado que contienen información actualizada de eventos recientes, por la rapidez en la propagación de la información y además por la confianza social (con el círculo cercano).

A pesar del potencial que tiene la información de las redes sociales para tareas de QA, no es sencillo utilizarla través de la aplicación de técnicas ya existentes basadas en sitios tradicionales de QA. Existen características únicas que diferencian el contenido social de las plataformas tradicionales de QA, por ejemplo; el tamaño de los mensajes suele ser corto y sin mucho contenido, por lo tanto no es simple determinar la calidad del mismo. Además el contenido es más propenso a ser ruidoso, irrelevante o por debajo del estándar. Lo anterior, en combinación con la alta cantidad de información que se genera en cada instante, constituye una sobrecarga de información que es compleja de extraer y estructurar para tareas de QA.

En esta tesis, estudiamos el potencial que tienen las plataformas de microblogs para tareas de QA. Además estudiamos las características que poseen las respuestas más relevantes dada una pregunta inicial. En particular, creamos un modelo de documento a nivel de hilos de conversación en microblogs que nos permitió agregar información de contenido y entrenar un modelo de ranking basado en preguntas y respuestas de tipo factoid. Nuestros resultados experimentales, llevados a cabo en Twitter, nos indican que esta plataforma social sí provee de información valiosa para tareas de QA. Además identificamos las principales características de contenido y sociales (y la combinación entre ellas) que permiten obtener respuestas relevantes. El modelo consigue una mejora de alrededor de un 62% con respecto al método base empleado. De hecho, empleando el mismo modelo ya entrenado y la técnica de Transfer Learning, fuimos capaces de responder preguntas más complejas de tipo non-factoid capturadas directamente de Twitter.

Abstract

Online social networks have changed how users seek and find information on the Web. In particular, community Question Answering (cQA) sites have emerged as platforms designed specifically for the exchange of questions and answers among communities of users. Although users tend to find good quality answers in cQA sites, there is evidence that users also engage in a significant volume of QA interactions in other types of social networking sites, such as Twitter. Research indicates that users opt for these non-specific QA social networks because not only they contain up-to-date information on current events, but also due to their rapid information propagation, and social trust.

Despite the potential of social media information for QA tasks, it is not straightforward to use this information through the application of existing techniques based on cQA sites. There are unique characteristics that differentiate social content from traditional cQA platforms; user messages can be short –without much content to determine their quality, as occurs in microblog platforms such as Twitter- and noisy, where content can be irrelevant or below standard. This, in combination with the high arrival-rate of new messages, constitutes an overload of information that makes it very complex to leverage this data for QA.

In this thesis, we address this problem by studying the feasibility of a microblog social media platform for QA, and the discriminating features that identify relevant answers to a particular query. In particular, we create a document model at conversation-thread level, which enables us to aggregate information and set up a learning-to-rank framework, using factoid QA as a proxy task. Our experimental results on Twitter data show that we can indeed use it as a QA retrieval resource. We are able to identify the importance of different features and combinations thereof that better account for improving QA ranking, achieving an MRR of 0.7795 (improving it by 62% over our baseline method). In addition, we provide evidence that our method allows to retrieve complex answers to non-factoid questions.

A mi familia, de siempre y la que estamos construyendo...

Agradecimientos

Primero que todo agradecer a mi esposa Erika, mi hija Maite y a mi otro hijo que viene en camino, por ser mi cable a tierra y la principal motivación para terminar mis estudios.

A Bárbara por su paciencia y apoyo en momentos que lo necesité durante este proceso. Gracias por tu trato, simpatía, consejos, ayuda y por considerarme en otros proyectos. A Denis por tener el tiempo de recibirme miles de veces en la PUC, por sus consejos, por las largas conversaciones de cualquier cosa y por su ayuda en mi investigación.

También agradecer a muchas personas que fueron parte de este proceso. Fueron fundamentales en cuanto a mantener un buen ambiente de trabajo y además por las buenas y largas conversaciones de cualquier cosa; Vanessa, Héctor, Cristobal, Miguel, Hernán, entre otros. Se me vienen a la mente muchas otras personas que por diversas razones personales, familiares o académicas no pudieron continuar sus estudios. Ellos también fueron parte importante de esto.

A Sandra y Angélica, gracias por su cordialidad y disposición para resolver cualquier eventualidad.

To Renato, for the long conversations about music, bands, guitars, etc. For his kindness and the continuous support of my English skills. Thanks for all and I hope this paragraph is well written. :)

Al DCC, por su buen ambiente de trabajo y por el financiamiento otorgado para algunas conferencias. A los directores del programa; Prof. Eric Tanter y Prof. Gonzalo Navarro, ambos siempre disponibles y dispuestos a solucionar cualquier inconveniente.

Al Instituto Milenio Fundamento de los datos (IMFD) y al antiguo Centro de Investigación de la Web Semántica (CIWS) por su constante apoyo en todas las actividades de mi doctorado.

Finalmente agradecer a CONICYT, por el financiamiento de estos estudios de doctorado a través de la beca de Doctorado Nacional 2013 Folio 21130931.

Gracias, totales!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Online social networking platforms have changed how people produce and consume Web content. Social networking sites are designed to facilitate interaction among users, allowing the creation of communities with different purposes. In particular, *community Question Answering (cQA)* Web sites are platforms that specialize in connecting users that are interested in expressing information needs in the form of questions, with other users that can provide answers to these questions. Examples of such sites are Yahoo! Answers[1], Stack Exchange[2], among others. Here, people can ask two kinds of questions: *factoid* and *non-factoid* questions. The first type requires just one answer or statement of fact, for example, "*What is the capital of France?*" The second one requires experiences, opinions, lists, recommendations or advice, for example, "*What are the best German beers?*" The resulting interactions among users are traditionally preserved permanently in the cQA Web site, constituting an historical knowledge base.

Microblog platforms, such as Twitter[3], allow users to exchange vast amounts of information in short messages (called *tweets*). Users of microblog platforms post real-time status updates and information on diverse topics. These social networks have had wide adoption, drastically increasing the volume of content published daily on the Web. Because the platforms are built to allow real-time updates, the content is typically short-lived and continuously replaced by newly arriving information. This information is considerably valuable, given that queries phrased as questions often represent complex information needs, which are not easily satisfied with traditional Web information retrieval techniques [79, 91].

The sustained use of Twitter for QA suggests that valuable historic information for automatic QA retrieval could be obtained from this platform. In fact, it can be considered as a way to complement information on traditional cQA Web sites, with more up-to-date and context rich answers.

On the other hand, analysis of Twitter shows that users are engaging in QA related conversations on a regular basis [48, 80, 121], constituting about 10% of the total messages.

[1] http://answers.yahoo.com

[2] http://stackexchange.com

[3] http://www.twitter.com

(1)	21:11:31	BREAKING: Lockdown in east #London after 'disturbance' - several injured including police https://t.co/nCmVlkjnP5 #BarkingRoad
(2)	21:19:28	@SkyNewsBreak any news re this apparent incident near London Bridge ?
(3)	21:19:52	Am on London Bridge. Something serious happening.
(4)	21:21:21	Just finished gig in Hays Galleria in #LondonBridge & there are serious amount of police & ambulances. Anyone know what's going on? #Borough
(5)	21:21:47	London Bridge is partly blocked southbound due to a collision. Traffic is slow on approach. Use other crossings where possible.
(6)	21:24:29	Another cab driver: man crashed into people then there was stabbing attack. London Bridge.
(7)	21:24:44	HUGE number of armed police gathered at London Bridge Crossroads, telling people to move away #LondonBridge @BBCBreaking
(8)	21:24:51	Very worrying reports of horrific incident, possible terror attack, on London Bridge.
(9)	21:24:59	Van drove into people on London Bridge, this is happening now. Really hope all are ok. We are in the middle of it. #londonbridge #london pic.twitter.com/FE0Oxi0YSD.
(10)	21:25:19	London Bridge?
(11)	21:25:19	Yes looks like something very serious happening on London Bridge.
(12)	21:25:32	@bbc5live what's happening at London Bridge?

Table 1.1: Chronology of the first tweets of London Attack taken from Twitter API between 21:06 UTC and 21:25 UTC (ie., 19 minutes). Notice question (12) -in yellow- could be answered by tweets (3), (5), (6), (7), (8) and/or (9) -in green-.

This behavior seems counterintuitive if we consider that there are other platforms specifically designed for QA. One vital aspect of the interaction is that answers can be obtained from a trusted source: the user's personal network of people they interact with on a daily basis [80, 98]. Another important aspect is the immediacy of information of user interactions on Twitter. Both of these drivers imply that answers to users questions will have temporal, social, and possibly geographical context.

The immediacy of Twitter offers a remarkable advantage over other platforms. This immediacy is helpful for answering time-sensitive questions that arise in emergency situations, for example, an earthquake or a terrorist attack. These kinds of events generate thousands of tweets posted per hour and many of them are questions. For example, Table 1.1 shows the first 20 minutes tweets regarding London Attack occurred on June 3, 2017 21:06 UCT. The first questions appear in (2), (4), (10) and (12), meaning 13-19 minutes after the event occurred. The rest of the tweets could be useful for understanding the event and answering some subsequent questions.

We inspect another event; the Westminster attack, which occurred on March 22, 2017 14:40 UCT. We analyze the tweets of the first 30 minutes after the event occurred. Table 1.2 shows some of these tweets. Surprisingly, the first related tweet was published just 1:48 minutes after the event occurs, and there were around 20 tweets in the first 5 minutes (likely, because it is a very popular touristic area). Notice that tweets are explicit about the description of the place and what happened. For instance, tweets from (1) to (7) contain explicit keywords about the place and the type of event that occurred: *Westminster*, *parliament*, *bridge*, *knife*, *police shot*, *attack*, *gunfire*, amount others. We also realized that some questions appear 12 minutes after the event occurred (see, (11) to (14)). In fact, using just the first 100 tweets yielded in the first 15 min, we could answer Twitter questions such as:

- *what's happening in the parliament?*
- *attack at Westminster?*
- *shots at parliament?*
- *parliament attacked?*

(1)	14:41:48	Gun shots shots outside Parliament now.
(2)	14:41:50	Shots fired outside parliament.
(3)	14:41:57	Sounds like gunfire at Parliament.
(4)	14:42:43	Just seen police shoot a man outside parliament, appeared to be wielding a knife
(5)	14:43:11	Police have just shot a man who was attacking officers at parliament's entrance.
(6)	14:43:15	Shots in Parliament Square. People running like crazy.
(7)	14:44:36	Man receiving medical attention outside Westminster Hall. I heard loud explosion on Westminster bridge. Public running away.
(8)	14:45:20	BREAKING Reports of gunfire outside Houses of Parliament in London. #Parliament https://t.co/nWpSOdk4fK
(9)	14:46:02	RT @TomMcTague: Someone shot outside gates of Parliament. Very serious situation. Attack on policeman.
(10)	14:46:25	Terrifying in #Westminster now. We have been told we can't leave our office. It looks like someone has been shot. Public ran in panic.
		...
(11)	14:52:47	Shooting in London near/on Westminster bridge, 6 wounded? #uk
(12)	14:54:25	Parliament attacked?
(13)	14:55:33	Guns in the Houses of Parliament ? Really ?
(14)	15:03:13	OMG what is happening in Europe? firing outside UK parliament, 13 injured. what kind of phobia is this?

Table 1.2: A sample of the first tweets of the Westminster attack taken from Twitter API between 14:40 UTC and 15:10 UTC (ie., 30 minutes). Notice that the first tweet appears 1 min 48 secs after the event. The yellow section highlights users' questions about the event.

- *gunfire at parliament?*
- *explosion in London?*
- *what's going on in London?*

Twitter has likely been one of the first platforms with this up-to-date information. Therefore, immediacy is an important advantage of Twitter because it allows us access to real-time information. In fact, it can be interesting to investigate the questions that arise in these critical events.

In this thesis, we investigate the feasibility of using microblog social media platforms as a QA resource. We create a ranking model for relevant answers and also identify the most important features for effective microblog QA. In short, we demonstrate that microblogs stores valuable information that we can preserve to satisfy a similar information need in the future. For these purposes, we use the microblog Twitter[4].

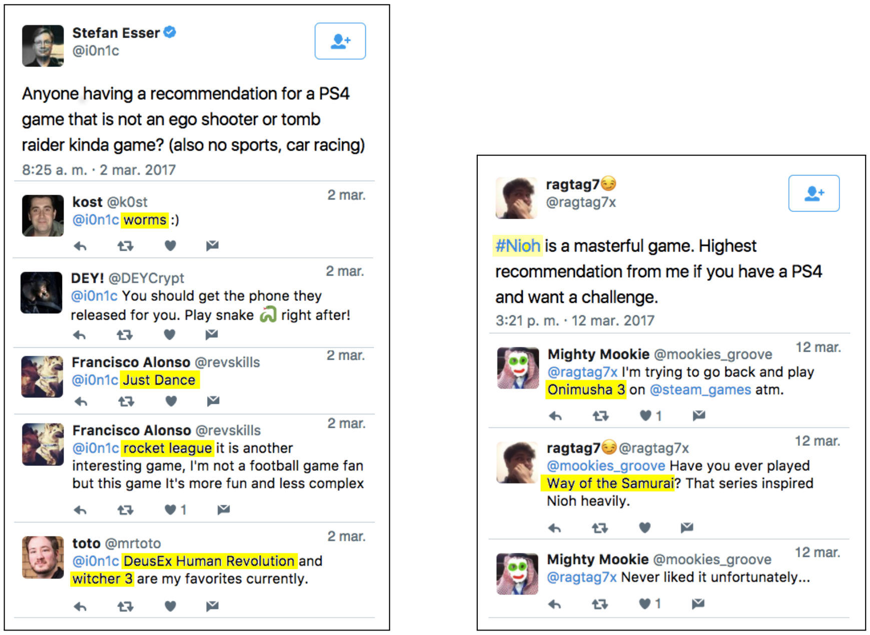
We approach this problem by proposing an aggregated document model that considers complete conversation threads on Twitter as documents, as opposed to single tweets. Lin et al. [69] report that users prefer answers embedded (in its original context) in a paragraph instead of just one exact answer. These answers tend to be more complete and also the number of related queries can decrease. For example, given the question: *“Who wrote "The Strange Case of Dr Jekyll and Mr Hyde"?’*[5], one candidate answer could be:

```
Robert Louis Stevenson, born on Nov 13,1850, wrote The Strange Case of
Dr. Jekyll and Mr. Hyde & Treasure Island. He had lustrous brown eyes.
```

This phrase shows the correct answer and additional information such as the birth date and

[4] http://www.twitter.com; Twitter is one of the most used microblog.

[5] Extracted from Twitter.



Stefan Esser @i0n1c

Anyone having a recommendation for a PS4 game that is not an ego shooter or tomb raider kinda game? (also no sports, car racing)

8:25 a. m. · 2 mar. 2017

kost @k0st · 2 mar.
@i0n1c worms :)

DEY! @DEYCrypt · 2 mar.
@i0n1c You should get the phone they released for you. Play snake 🐍 right after!

Francisco Alonso @revskills · 2 mar.
@i0n1c Just Dance

Francisco Alonso @revskills · 2 mar.
@i0n1c rocket league it is another interesting game, I'm not a football game fan but this game It's more fun and less complex
♥ 1

toto @mrtoto · 2 mar.
@i0n1c DeusEx Human Revolution and witcher 3 are my favorites currently.

(a) Conversation thread 1.

ragtag7😏 @ragtag7x

#Nioh is a masterful game. Highest recommendation from me if you have a PS4 and want a challenge.

3:21 p. m. · 12 mar. 2017

Mighty Mookie @mookies_groove · 12 mar.
@ragtag7x I'm trying to go back and play Onimusha 3 on @steam_games atm.
♥ 1

ragtag7😏 @ragtag7x · 12 mar.
@mookies_groove Have you ever played Way of the Samurai? That series inspired Nioh heavily.

Mighty Mookie @mookies_groove · 12 mar.
@ragtag7x Never liked it unfortunately...
♥ 1

(b) Conversation thread 2.

Figure 1.1: Example of two conversation threads extracted from Twitter that can answer the question: *"What are good games for ps4?"* (ps4: playstation 4). The initial tweets are paraphrased as questions followed by replies. User suggestions are highlighted in yellow.

another book from the author. It is common to find these kinds of answers in QA platforms, but not in Twitter (or in microblogs). In fact, the problem is even worse since messages are very short and they do not provide much information. However, people use Twitter to ask certain types of questions mainly for the rapidness in which they obtain answers. Therefore, we propose to use conversation threads instead of single tweets to obtain more and, likely, better information [69].

Figure 1.1 shows examples of two conversation threads yielded on Twitter and related to the question "*What are good games for ps4?*". In this example, each thread provides several relevant answers for the question, which can only be identified by reviewing the complete conversation. This approach also has the potential of providing complex answers to information needs expressed as questions, based on the diversity of information found in a particular thread or by aggregating multiple conversations.

To address this task, we first performed an exploratory analysis of questions and answers yielded in Twitter. We address the task of creating a method for retrieving the most relevant historical conversation threads that can answer a given query q by leveraging Twitter as a QA repository. In particular, we study how the combination of questions, replies and features (such as *retweets*, *replies*, *favorites*, etc.) can be useful in identifying whether a conversation thread triggered by a question is relevant in terms of information quality, to the particular conversation topic. For this reason, (1) we use a thread-level document representation of microblog data to get more information from tweets. We then (2) perform an analysis using well known datasets of one type of question: factoid questions. We studied the most important features to find relevant answers and create a ranking model based on several of these features. The next step (3) is to study if this approach can perform well with more complex types of questions: non-factoid QA. For this task, we conducted a manual evaluation of non-factoid questions related to recommendations (Figure 1.1 shows two examples). The results indicate that our approach is a very good starting point for answering more complex and context rich questions.

In particular, we observe several benefits from this proposal:

1. It can allow people to satisfy their information needs quickly, especially in time-sensitive situations.
2. It is a tool to deal with part of the information overload problem in streaming platforms.
3. Historical data can help to satisfy similar information needs that may emerge in the future.
4. Users could make use of answers provided by the crowd – even from people not in their direct social circles.

To the best of our knowledge, there are no methods that leverage data from streaming platforms for QA tasks.

We therefore define the goals of this thesis as follows:

General Objective. The main goal is to identify useful and high-quality information in microblog platforms that is relevant for QA purposes, which can be used in other applications and/or to deal with future information needs from users.

Hypothesis. There is information in microblog platforms that allows us to effectively satisfy information needs related to QA.

Specific Objectives. We focus on the following specific objectives:

1. Demonstrate empirically that microblogs can be a source of relevant information for QA.
2. Design a method for retrieving and extracting relevant features from microblogs.
3. Identify the most important features for effective microblog QA using microblog data.
4. Create a ranking model that satisfies information needs related to QA using Twitter data.

Research Questions. We define three main research questions for this thesis:

- **RQ1:** Is it possible to retrieve relevant answers to incoming questions using historic Twitter data?
- **RQ2:** Which features are the most important for finding relevant answers to questions using Twitter data? Are these features the same as in traditional cQA platforms?

- **RQ3:** Can Twitter conversation threads be used to answer questions (as opposed to using single tweets) in order to provide answers to complex questions?

Contributions. The main contributions of this work are:

- A model to find candidate answers for questions using microblog data, and rank these answers based on their relevance to the query.
- A study about the most relevant features for ranking.

Our experimental results show that by using a document representation of whole conversation threads rather than single tweets, and that by incorporating content-based and social-based features to answer questions, the ranking improves significantly in all our evaluation metrics. However, we note that the real value of our work lies in proposing the social platform Twitter for specific QA, as a source for answering question and in identifying which features contribute the most to finding relevant answers.

This thesis is organized as follows: Chapter 2 describes relevant concepts and techniques necessary to understand this document; Chapter 3 presents the current state-of-the-art and its relation to our work; Chapter 4 describes a preliminary exploratory analysis of relevant information extraction from social media for QA purposes; Chapter 5 propose a model to address this problem as a ranking problem; Chapter 6 presents our model validation; Chapter 7 is a empirical study of complex questions (non-factoid) using the proposed model; Chapter 8 presents the limitations of this work and overall discussion; and finally Chapter 9 are the conclusions and future work.

The research of this thesis generated three publications: two accepted and one under review.

- [44]; Jose Herrera, Barbara Poblete and Denis Parra. Retrieving Relevant Conversation for Q&A on Twitter. In proceedings of International Workshop on Social Personalisation & Search (SPS 2015)
- [45]; Jose Herrera, Barbara Poblete and Denis Parra. Learning to Leverage Microblog Data for QA. In Proceeding of European Conference on Information Retrieval 2018.
- Jose Herrera, Denis Parra and Barbara Poblete. Social QA in non-CQA platforms. `[Under review]` 2019.

Chapter 2

Background

This chapter describes in depth, the terms and concepts that we consider relevant for a better understanding of this document.

2.1 Natural Language Processing (NLP)

NLP is a set of techniques widely used by search engines and voice interfaces to analyze and recognize speech. It is a way for machines to analyze, understand, and derive meaning from human language in a smart and useful way. It is used for machine translation such as text mining and automated Question Answering. Developers can perform tasks such as translation, automatic summarization, named entity recognition (NER), sentiment analysis, speech recognition, topic segmentation and part-of-speech tagging.

In this thesis we extract content-based features from tweets using NLP techniques.

2.1.1 Part-of-Speech (POS)

The part-of-speech tagging –also known as POS, word classes or syntactic categories– identifies syntactical function of words. It determines if a word is a noun, a verb, a conjunction, etc. POS tells us a lot about likely neighboring words, for example, that nouns are preceded by determiners and adjectives or verbs by nouns. Hence, we can understand more clearly the syntactic structure around the words.

Jurafsky and Martin [53] divided POS in two categories: **Closed classes** and **Open classes**. Closed classes cannot change in the future or have new words added, for example, prepositions. These kinds of words tend to be short and occur frequently in any dataset. In contrast, open classes could change, remove or add new ones, for example, *iphone* and *fax* are new nouns. In our study, we concentrate on Closed Classes.

Category	sub category	Refer to	Examples
Noun	Common nouns	Generic names of things.	cat, car, desk.
	Proper Nouns	Name of persons, places or things.	Messi, Paris, Colorado.
	Proper Concrete	Can be perceived through your 5 senses.	Sand, flower, car
	Proper Abstract	Cannot be perceived through your 5 senses.	Happiness, bravery
	Proper Count	Anything countable.	Ball, kitten.
	Proper Mass	Non-countable things.	Kilo, meters.
	Proper Collective	Group of persons, animals, things.	Class, faculty.
Pronoun		Replacement for nouns.	He, she, it, his, her.
Adjective		Describes a noun or a pronoun.	The dog is *Huge*, I have *two* cats.
Verb		Shows an action.	Was, were, start
Adverb	Manner	How something happens or an action is done.	Annie danced gracefully.
	Time	When something happens or when it is done.	She came yesterday.
	Place	Where something happens or where something is done.	Of course, I looked everywhere!
	Degree	Intensity or the degree of something.	The child is very talented.
Preposition		Words that specify location or a location in time.	Above, below, before, near, since.
Conjunction		Joins words, phrases, or clauses together.	And, yet, but, for, nor, or, and so.
Interjection		Words which express emotions.	Hey!, Hurray!, Ouch!

Table 2.1: Parts-of-speech categories. Part of this table is available at http://partofspeech.org.

In **closed clases** we consider four major types: **nouns**, **verbs**, **adjectives** and **adverbs**.

- **Nouns.** They are words related to people, places or things. For example, *car*, *dog*, even verb-like terms like *pacing* (*"His pacing to and fro became quite annoying"*). Usually nouns occurs with determiners; *its bandwidth*, *UEFA's cup*. Moreover, nouns fall into two classes:
  - **Proper nouns**. They are names or entities such as; *Jose*, *Chile*, *Apple*. These kinds of words are usually capitalized.
  - **Common nouns.** They are divided in **count nouns** and **mass counts**. The first one occurs in singular and plural, and they can be enumerated: *one cat/two cats*, *one chair/two chairs*. In contrast, mass nouns cannot be enumerated; *two snows*.
- **Verbs.** They are words that refer to actions and processes; *go, be, write*. Moreover, verbs have inflections because of the different verbal tenses; *go, went, gone*.
- **Adjectives.** They include properties or qualities; *white*, *old*, *good*.
- **Adverbs.** They are words that complement a verb, adjective or other adverbs; *"the pencil is here on my desk"* or *"the transaction was extremely slow"*. However, there are several categories in adverbs such as: locative adverbs, temporal adverbs, etc.

This four types are the most basic categories of words. Table 2.1 summarizes these four basic POS classes and adds others.

In this thesis, we use POS tagging to assign a label to each word of a tweet that is part of a document. Also, given that we study microblogs, we use a special POS tagging for Twitter text made by Gimpel et al. [39]. They developed a POS tagset for Twitter and manually tagged more than 1,000 tweets, reaching an accuracy of about 90%. Moreover, the tool includes social tags based on Twitter features such as hashtag, mentions, among others. Table 2.2 shows the complete list of POS tags for Twitter.

Tag id	Description
1	Common noun
2	Pronoun (personal/WH; not possessive)
3	Proper noun
4	Nominal + possessive
5	Proper noun + possessive
6	Verb including copula, auxiliaries
7	Nominal + verbal (e.g. I'm), verbal + nominal (let's)
8	Proper noun + verbal
9	Adjective
10	Adverb
11	Interjection
12	Determiner
13	Pre- or postposition, or subordinating conjunction
14	Coordinating conjunction
15	Verb particle X existential there, predeterminers
16	X + verbal # hashtag (indicates topic/category for tweet)
17	@ at-mention (indicates a user as a recipient of a tweet)
18	Discourse marker, indications of continuation across multiple tweets
19	URL or email address
20	Emoticon
21	Numeral
22	Punctuation
23	Other abbreviations, foreign words, possessive endings, symbols, garbage

Table 2.2: POS tagset for Twitter (called TweetNLP) introduced by Gimpel et al [39]. Tags 17-20 are social features of Twitter.

2.1.2 N-grams

N-grams tagging is an essential subtask of speech recognition that allows us to predict the next words of a text. For instance, the text: *I am leaving in about fifteen...*, could be followed by several words such as: *minutes*, *seconds*, *days*, etc.

In general, the models of word sequences are probabilistic models that assign probabilities to strings of words, whether for computing the probability of an entire sentence or for a probabilistic prediction of what the next word will be in a sequence [53].

A way to compute the probability of a complete string of words represented as $w_1, w_2, w_3, ..., w_n$ or w_1^n is:

$$P(w_1, w_2, w_3, ..., w_{n-1}, w_n) \;\; or \;\; P(w_1^n) \tag{2.1}$$

We can decompose it as:

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)...P(w_n|w_1^{n-1}) = \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \tag{2.2}$$

But, it is not easy to compute $P(w_n|w_1^{n-1})$, because we need to calculate the probability of a word given a long sequence of preceding words. Hence, we can apply an approximation of a word given the previous word using the **Markov** assumption. The **Markov** assumption allows us to predict a future model without looking too far in the past [53]. For example, the *bigram* model approximates the probability of a word given all previous words $P(w_n|w_1^{n-1})$ using the conditional probability of the preceding word $P(w_n|w_{n-1})$, which means that instead of computing this probability:

$$P(\textit{minutes} \mid \textit{I am leaving in about fifteen}) \tag{2.3}$$

we can approximate to:

$$P(\textit{minutes} \mid \textit{fifteen}) \tag{2.4}$$

Of course, we can observe more than one word (or token) of the past. In these cases we use bigrams, trigrams and so on. Hence, *N-grams* is the $N-1$ order of the Markov model, where N is the amount of words that we look back on.

Summarizing, to compute *N-grams*, we use the conditional probability of the next word:

$$P(w_n|_1^{n-1}) \approx P(w_n|w_{n-N+1}^{n+1}) \tag{2.5}$$

In this thesis, we compute unigrams, bigrams and trigrams from tweets.

2.2 Learning to Rank (LTR)

Everyday, the Web generates more information than humans can gather. Hence, it is necessary to have a method that can filter this information based on the ranked relevance of documents. One method that addresses this issue is called **Learning to Rank** (LTR). LTR uses individual documents as an input, then compute a score measuring the matching between potentially relevant documents and the query. Then all the documents are sorted in descending order of their scores. In this thesis, we use this technique to rank answers of conversation threads from Twitter.

LTR is widely used in several types of ranking problems in information retrieval (including traditional QA), natural language processing (NLP) and recommender systems such as [1, 26, 27, 30, 104, 105, 120]. The main goal is to produce a ranked list of documents according to the relevance between these documents and a query. The relevance refers to documents that have a relation to the query. For example, for the query "question answering problem" posed in a search engine, the relevance is given by web documents that address question answering problems. Other topics are considered as irrelevant documents.

The methodology to address a ranking problem is slightly different from the Information Retrieval (IR) process. In 2001, Voorhees [109] defines a methodology based on sharable document collection, queries and relevance assessments. Applying this paradigm to LTR [70], the complete process is:

1. Collect a large number of queries.
2. For each query:
   - Collect documents $\{\mathrm{d}_j\}_{j=1}^{m}$ associated with the query.

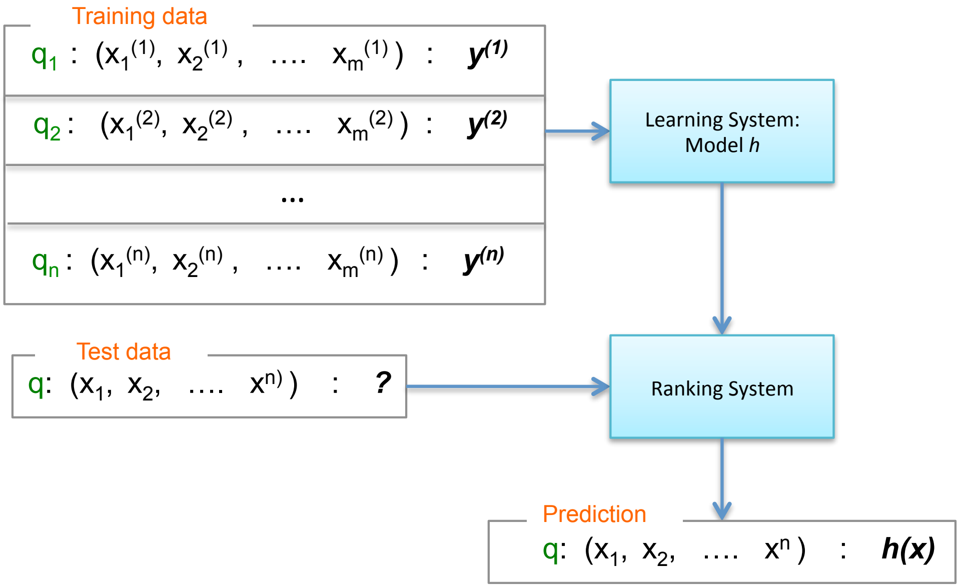

Training data
q1 : (x1(1), x2(1) , ... xm(1)) : y(1)
q2 : (x1(2), x2(2) , ... xm(2)) : y(2)
...
qn : (x1(n), x2(n) , ... xm(n)) : y(n)
Learning System:
Model h
Test data
q: (x1, x2, ... xn)) : ?
Ranking System
Prediction
q: (x1, x2, ... xn) : h(x)

Figure 2.1: Learning to Rank framework. The diagram desing was inspired in Liu et al. [70]

- Obtain the relevance of each document (human evaluation).
- Learn a ranking model (using a ranking algorithm).
- Measure the difference between the ranking results and relevance judgment. The model is adjusted in each iteration.

3. Use the average measure on all the queries in the test set to evaluate the performance of the ranking model.

Since LTR is a supervised learning technique, a training set is needed. A typical training set consists of n training queries q_i and their associated documents represented by feature vectors. Figure 2.1 shows the complete framework of LTR proposed by Liu [70], where each query is associated with a set of documents with relevance judgments. A query q_i has m features, $x^{(j)}$ are the feature vectors of the query and $y^{(\mathrm{i})}$ is the relevance of q_i. A model h is learned with training data using a specific ranking algorithm. Finally, given test data, the ranking value is predicted using the model $h(x)$.

2.2.1 Pointwise, Pairwise and Listwise Approaches

There are three different approaches of LTR. They define different input and output spaces, and also use different hypotheses and loss functions.

- **The Pointwise Approach**. In this approach, the ranking problem is transformed to a classification problem. It means that each document is considered as an individual instance to learn and the group structure of ranking is ignored. In particular, the *input space* contains a feature vector of each single document. The *output space* contains the relevance degree of each single document. Some methods are: Subset Ranking [24], McRank [68] and Prank [25].

- **The Pairwise Approach**. This approach transforms a ranking problem in a pairwise classification or pairwise regression. Document pairs are now taken as instances in learning. The group structure of ranking is also ignored. In particular, the *input space* consists in pairs of documents represented by feature vectors. The *output space* contains the pairwise preference in binary form $(1, -1)$ between each pair of documents. Some methods are: Ranking SVM [43], RankBoost [34] and RankNet [17].
- **The Listwise Approach**. This approach takes a list of documents associated with a query as instances. It takes ranking lists as instances in both learning and prediction. In this case, the group structure of ranking is now considered. In particular, the *input space* contains a set of documents associated with a query. The *output space* contains the ranked list. Some methods are: ListNet [18], ListMLE [112] and AdaRank [113].

2.2.2 LTR Methods

There are several types of LTR methods [70]. In this thesis we use four pairwise models: MART [36], Ranknet [17], Rankboost [34] and LambdaMart [16]. These methods are well-used in similar related studies. They are defined as follows:

- **Multiple Additive Regression Trees (MART) [36, 70]**. Also called *Gradient Boosted Regression Trees* (GBRT). MART is a pairwise method based on trees. It is a boosting algorithm tree model in which the output of the model is a linear combination of the outputs of a set of regression trees. It can be viewed as a method that performs gradient descent in a function space, using regression trees.

  The loss function used is:

$$L(f; x_u, x_v, y_{u,v}) = \frac{1}{2}(max\{0, y_{u,v}(f(x_v) - f(x_u)) + \tau)\})^2 - \lambda\tau^2 \tag{2.6}$$

  where, f is the scoring function, x_u and x_v are two documents associated to the query q. $y_{u,v}$ is the ground truth that indicates that document x_u should be ranked before document x_v (i.e., $y_{u_v} = 1$). If the predicted preference by the scoring function f is equal to the ground truth label $y_{u,v}$, the loss function will be zero. τ and λ are regularization factors which avoid obtaining a constant optimal scoring function.
- **Ranknet [17, 70]**. Ranknet is a pairwise method which trains a neural network with gradient descent to obtain a ranking function. By default, it uses a three-layered neural network with a single output node to compare pairs. Ranknet optimizes for (a smooth, convex approximation to) the number of pairwise errors [16].

  The lost function is defined as: given two documents x_u and x_v associated to query q, a target probability $\bar{P}_{u,v}$ is constructed based on ground truth labels.

$$P_{u,v}(f) = \frac{exp(f(x_u) - f(x_v))}{1 + exp(f(x_u) - f(x_v))} \tag{2.7}$$

  Therefore, the loss function is:

$$L(f; x_u, x_v, y_{u,v}) = -\bar{P}_{u,v}\ log\ P_{u,v}(f) - (1 - \bar{P}_{u,v})\ log\ (1 - P_{u,v}(f)) \tag{2.8}$$

where, f is the scoring function, x_u and x_v are two documents associated to the query q and $y_{u,v}$ is the ground truth that indicates that document x_u should be ranked before document x_v (i.e., $y_{u_v} = 1$). $\bar{P}_{u,v}$ is defined above.

- **RankBoost [34, 70]**. Rankboost is a pairwise method that solves the preference learning problem. It combines weak rankers in several iterations and learns the optimal weak ranker based on a distribution of the document pairs. It uses a linear combination to get the final ranking function, inspired by AdaBoost [35] for classification over document pairs (Rankboost is defined on document pairs unlike Adaboost which is defined on individual documents).

  The function minimizes the exponential loss defined as:

$$L(f; x_u, x_v, y_{u,v}) = -exp(-y_{u,v}(f(x_u) - f(x_v))) \tag{2.9}$$

  where, f is the scoring function, x_u and x_v are two documents associated to the query q and $y_{u,v}$ is the ground truth that indicates that document x_u should be ranked before document x_v (i.e., $y_{u_v} = 1$).

- **LambdaMart [16]**. LambdaMart is a pairwise/listwise method that combines the approaches of MART [36] and LambdaRank [70]. The latter only need the gradients of the costs with respect to the model scores to train a model. In contrast, LambdaRank uses gradient boosted decision trees and a cost function derived from LambdaRank for solving a ranking task. LambdaMart has been shown to have better results than LambdaRank [16].

2.2.3 Evaluation metrics

Evaluation plays an important role in information retrieval. We list the most popular metrics for evaluating rankings as most widely used in LTR studies.

- **Mean Reciprocal Rank (MRR)** : The MRR is the inverse of the position of the first relevant result. If there are no relevant result, the MRR is 0 [9]. Thus, MRR is calculated as:

$$MRR = \frac{1}{r_q}$$

  where r_q is the rank of the first relevant result. A high MRR implies that the ranking places the most relevant result near the top of the list.

  For example, for the following case the MRR is $\frac{1}{3} = 0.33$.

```
Instance 1 .... (non-relevant)
Instance 2 .... (non-relevant)
Instance 3 .... (relevant)
```

  For the following case the MRR is $\frac{1}{1} = 1$.

```
Instance 1 .... (relevant)
Instance 2 .... (non-relevant)
Instance 3 .... (relevant)
```

  For the following case the MRR is 0.

```
Instance 1 .... (non-relevant)
Instance 2 .... (non-relevant)
Instance 3 .... (non-relevant)
```

- **Normalized Discounted Cumulative Gain (nDCG)**: This metric measures the gain of a document discounted by the logarithm of its position. This accumulated gain is high when relevant elements appear at the top of the list and the non-relevant elements are placed at the bottom. To calculate $nDCG$, we must calculate the DCG of the ranking π and, then calculate the ideal DCG (iDCG) which represents the DCG considering that all the relevant results are at the top the list [9, 76].

  $$nDCG@k(\pi) = \frac{DCG@k(\pi)}{\mathrm{i}DCG@k(\pi)}$$

  where,

  $$DCG@k(\pi) = \sum_{\mathrm{i}=1}^{k} \frac{2^{R_{\mathrm{i}}-1}}{log_2(1+\mathrm{i})}$$

  R_{i} is the boolean relevance and, i and k are the number of instances.

In this thesis we use mainly these two metrics to evaluate several LTR models.

2.3 Word Embeddings

Word embeddings are a distributional representation where words with similar meaning have a similar representation. One of the most popular implementations is **word2vec** by Mikolov et al. [77]. It is a statistical method for efficiently learning a standalone word embedding from a text corpus. In this thesis we use word embeddings to compute different distance metrics between the query and threads. We also represent the initial questions and conversation threads for training purposes.

To build the embedding representation, word2vec trains a model using a corpus of billions of words to learn high-quality word vectors. With this representation, the similarity of word representations goes beyond simple syntactic regularities. For example, it is possible to address a single algebraic operation with words such as: vector(“King”) - vector(“Man”) + vector(“Woman”) in the text. The resulting vector is closest to the vector representation of the word *“Queen”*.

Two different learning models were introduced by Mikolov et al. [77] to learn embeddings; they are: **Continuous Bag-of-Words (CBOW)** and **Continuous Skip-Gram Model (Skip-gram)**. CBOW predicts the current word based on its context while the Skip-gram model predicts the surrounding words given a current word. Both learn words of their local context using a configurable window parameter of neighboring words. Figure 2.2 shows a diagram of these word2vec’s models.

The classic model, i.e., the vector space model, on the other hand, treat words as discrete and unique symbols. For instance, the word “cat” and “dog” could be represented as `id_100`

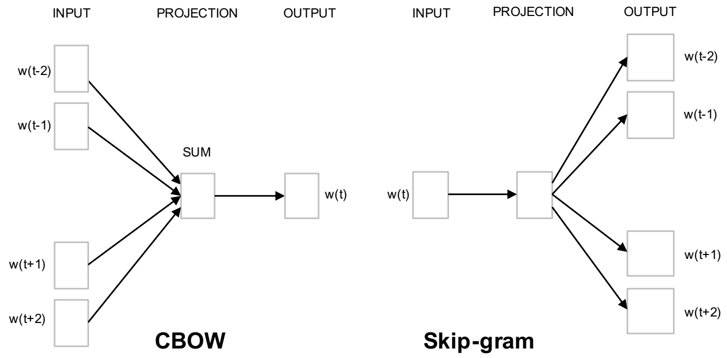


Figure 2.2: Word2vec's training Models. The diagram design was inspired in Mikolov et al. [77].

and `id_120`, respectively, but this representation does not consider the relationship of words. In contrast, word embeddings model the relation among words based on their context.

Several works in NLP [22, 58] have significantly improved their performance using word-vector embeddings such as word2vec [77] rather than the traditional vector space model with TF-IDF weights.

Chapter 3

Literature Review

Preamble

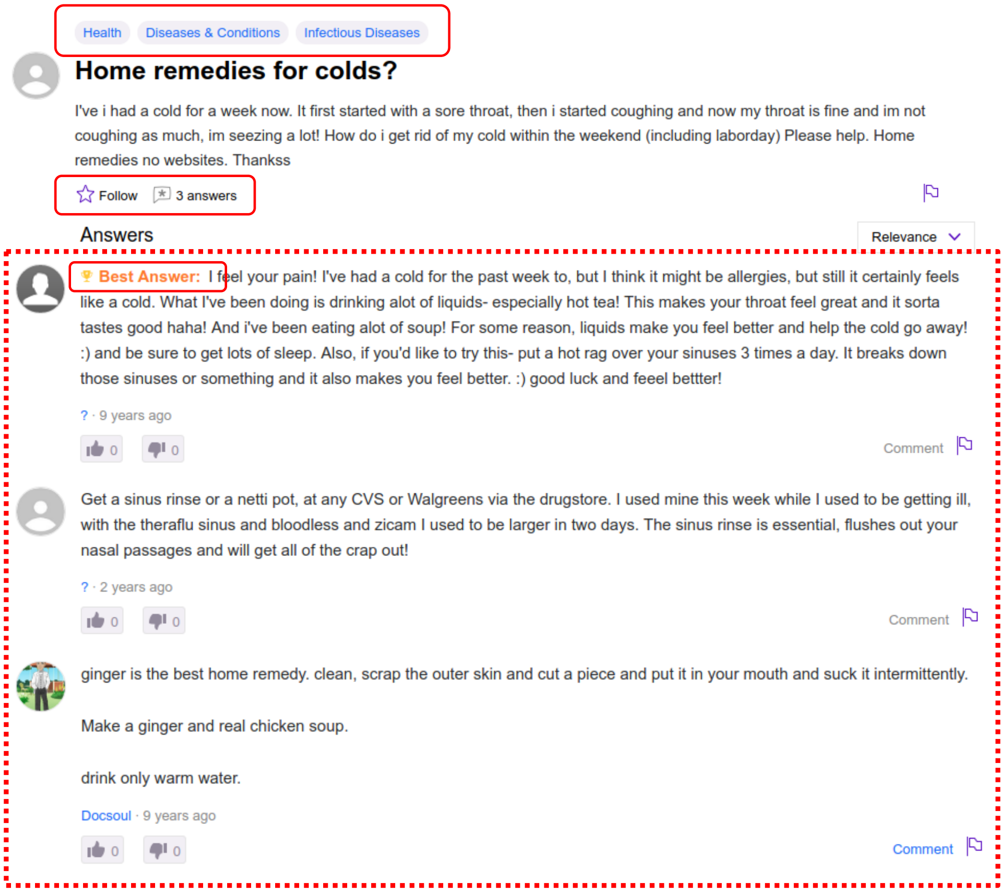
We found several QA studies which address traditional QA websites such as Yahoo! Answers and StackExchange. The availability of large public datasets of well-known QA websites has helped create different models that focus on, for example, predicting best answers, ranking answers, finding expertise, among others ([13, 83, 111, 114]). Likewise, social networks have been incorporated in the QA community research. However, studies in traditional QA are not fully reproducible in social network platforms. Hence, QA community research has studied new methods, new features and new techniques to reproduce some approaches for cQA.

In this chapter, we discuss the current state of the art for QA research. We first describe the QA task in online social networks. Then, focus on topics that are more relevant for our current work, as well as other QA topics that are useful to understand the QA problem.

3.1 Traditional QA

Traditional QA websites allow users to share knowledge through questions and answers. These sites provide an alternative channel for Web searches and they generate increasing amounts of rich online content. Users present detailed information needs and get direct responses authored by humans [2]. The main task of the QA problem is to automatically answer a question posed in natural language. The answer's extraction can use complex techniques such as Natural Language Processing (NLP).

The first QA services appeared about 30 years ago. Lots of them were developed as internal systems for companies. They later became known to the whole world through the Internet boom. Kolomiyets et al. [59] published an exhaustive history about these platforms in 1961.



Home remedies for colds?

I've i had a cold for a week now. It first started with a sore throat, then i started coughing and now my throat is fine and im not coughing as much, im seezing a lot! How do i get rid of my cold within the weekend (including laborday) Please help. Home remedies no websites. Thankss

Best Answer: I feel your pain! I've had a cold for the past week to, but I think it might be allergies, but still it certainly feels like a cold. What I've been doing is drinking alot of liquids- especially hot tea! This makes your throat feel great and it sorta tastes good haha! And i've been eating alot of soup! For some reason, liquids make you feel better and help the cold go away! :) and be sure to get lots of sleep. Also, if you'd like to try this- put a hot rag over your sinuses 3 times a day. It breaks down those sinuses or something and it also makes you feel better. :) good luck and feeel bettter!

? · 9 years ago

Get a sinus rinse or a netti pot, at any CVS or Walgreens via the drugstore. I used mine this week while I used to be getting ill, with the theraflu sinus and bloodless and zicam I used to be larger in two days. The sinus rinse is essential, flushes out your nasal passages and will get all of the crap out!

? · 2 years ago

ginger is the best home remedy. clean, scrap the outer skin and cut a piece and put it in your mouth and suck it intermittently.

Make a ginger and real chicken soup.

drink only warm water.

Docsoul · 9 years ago

Figure 3.1: A QA sample taken from Yahoo! Answers. Special features of these kinds of platforms are inside of the red rectangles. The dotted rectangle shows all the answers of these question and the first is the "best answer".

Today, the most popular cQA website is Yahoo! Answers[1]. It receives more than 800,000 questions and answers daily[2]. Other popular sites with high traffic are Baidu Zhidao[3] (Chinese) and Stack Overflow[4] (for language programming). These websites archive hundreds of millions of QA exchanges and continue receiving a large number of questions.

Figure 3.1 shows a QA sample taken from the Yahoo! Answers website. It has the following characteristics (top to bottom):

- question categorization (by tags),
- an option to follow questions (or save) and the number of answered question by user and,
- a best answer selection (chosen by users).

In addition, it is possible to get the user profile by placing the cursor over the profile image.

[1] https://answers.yahoo.com/

[2] Yahoo! Answers blog (may 2010), http://yanswersblog.com/index.php/archives/2010/05/03/1-billion-answers-served/

[3] https://zhidao.baidu.com/

[4] http://www.stackoverflow.com

In short, QA websites provide special features for QA interaction. Yahoo! Answers is one of the most complete in terms of characteristics and functionality. Over time, QA websites have become more interactive and user participation has increased, making them more dynamic. The generation of this information from users gives rise to interesting studies such as finding quality, special characteristics, patterns, expertise, authority, etc.

3.2 QA in Social Networks

Although QA sites are popular, users also ask a significant volume of questions online in other non-specialized, but more popular networking platforms such as Facebook or Twitter. This behavior might seem counterintuitive, especially for microblogs, due to the volatile nature of messages and the lack of special incentives to motivate users' answers that are available in QA sites, –such as badges and enhanced rights for active users. This seemingly suboptimal behavior might be explained by the observations of Morris et al. [80], who showed that the main motivation that users have for asking questions online on QA platforms is to receive quick and trustworthy answers – something that can be potentially achieved in a massive microblogging site like Twitter. In a preliminary analysis, we also found that around 10% of the Twitter stream corresponds to QA messages (similar measures were obtained by Efron et al. [32] and Li et al. [67]). Moreover, a rough inspection of QA conversation threads on Twitter reveals a high redundancy of questions over time, meaning that there is a high chance of finding answers to newly asked questions. This trend of QA usage in Twitter indicates an increasing potential for fulfilling current user information needs based on similar questions already answered in the past.

In this thesis we focus on QA studies using microblogs, in particular, Twitter. The majority of the studies that we inspected were addressed on Twitter, most likely because Twitter provides an API to get (a portion) of the streaming data.

3.2.1 Microblogs

Microblogging is an emerging form of communication. Users can publish brief message updates that can be submitted using several channel types such as those from the Web or mobiles, and also, in a variety of content formats including text, images, video, audio, and hyperlinks. The data are streaming quickly, hence messages can get lost (or get old) in the timeline after awhile. Examples of these kinds of platforms are: Twitter[5] (2006), Sina Weibo[6] (2009), Tumblr[7] (2007), Instagram[8] (2012), Vine[9] (2013), among others. We describe the first two.

[5] http://www.twitter.com

[6] http://www.weibo.com

[7] http://www.tumblr.com/

[8] http://www.instagram.com/

[9] http://vine.co/

Twitter. It is one of the most popular (and oldest) microblogging services. The messages are called *tweets* and allow up to 280 characters[10]. The timeline is a text stream in which users can post or read about real-time news, opinions, things, events of interest or share an image or video.

In the friendship interaction a user follows another in order to see that user's tweets, but it is not reciprocal. Hence, a user A has two lists: a following list (users followed by A) and a followers list (users who follow A). By default all tweets are public and there are five special social interaction features:

- **Favorites**: users can mark a tweet as "I like it".
- **Retweets**: to repost, share or repeat a tweet to the user's friendship network.
- **Reply**: to reply to a tweet and generate a conversation thread in one thread depth.
- **Mentions**: a user A can mention (or reply) to user B using @B.
- **Hashtags**: a keyword to describe a topic, for example, `#WorldCup`.

Sina Weibo. It is a popular Chinese microblog. It has the same Twitter features (favorites, retweets, reply, mentions, hashtags) and also the same friendship interaction (followers and followings). The only difference is in the messages length, because Sina Weibo allows messages up to 140 characters.

In the last year, Sina Weibo has slightly overtaken Twitter in user numbers and in market influence[11]. For example, in the first quarter of 2017, Sina Weibo had reported 340 million users and Twitter reported 328 million. Moreover, Sina Weibo has a market capitalization of US$16.4 billion vs US$13.5 billion of Twitter. However, both platforms are very similar in terms of user behavior and also have similar features for user interaction.

The main differences between microblogs versus traditional blogs are:

- Less time to consume content.
- The opportunity for users to produce more frequent posts.
- Freshness of the information (for example, for emergency situations).

Therefore, we reviewed similar studies addressing traditional QA platforms and applied some of those techniques to microblogs.

[10] In 2017, the message limit was increased from 140 to 280 characters.

[11] `http://www.forbes.com/`: The Rise Of Weibo: Lessons Twitter Can Learn From Chinese Upstart

3.3 Studies related to this thesis

3.3.1 Characterization of QA

In this thesis, we inspect and analyze content and social features of Microblogs. In general, we use state-of-the-art features and others that we ourselves extracted. We cover this issue with more details in Chapter 6. To the best of our knowledge, there are no methods that leverage information related to satisfying information needs in streaming platforms related to QA.

Identifying special characteristics or patterns in QA platforms are associated with prediction or classification purposes. For example, Cong et al. [23] detect and extract question-answer pairs inside threads of three well-known travel forums: TripAdvisor[12] and Lonely-Planet[13]. They inspect 40 forums and found that 90% of them contain QA knowledge. Through classification and a graph-based methods, they are able to detect special patterns to find question-answer pairs inside discussions. Amiri et al. [3] inspect questions related to opinions in Yahoo! Answers. They propose a dictionary of opinion words and special patterns to predict these types of questions.

Microblogs. There are also studies about characterization of QA in Social Networks. Morris et al. [80] perform a characterization study of questions asked in Facebook and Twitter, and found that the most asked questions are about recommendations, opinions and factual knowledge (Figure 7.1 features the top asked questions). Paul et al. [98] conducted a similar study in Twitter and found that the most popular questions are "rhetorical" and "factual Knowledge". Liu and Jansen [74] create a taxonomy for classifying questions that users ask in Twitter based on four criteria: accuracy, social, knowledge and conversational. Li et al. [67] detect questions in Twitter that need to be answered using content-based and social-based features. A similar study by Zhao and Mei [121] detects information needs in Twitter through state-of-the-art techniques and a comprehensive collection of other features. They observe that questions asked in Twitter are different from some topics being tweeted. In particular, they predict the trends of Google queries from information needs obtained on Twitter. It means that information needs detected in Twitter have a considerable power of predicting the trends of Google queries.

Another important characteristic of QA is finding similar questions. These kinds of studies are relevant because they can be helpful in answering other similar questions already answered previously. For example, Blooma et al. [12] address this problem in Yahoo! Answers. This is one of a few research projects that use clustering techniques to identify similar questions. Based on the relationship of questions with common answers and users (meaning questions, answers, askers and answerers), they create an algorithm to build a quadripartite graph considering these features. With an agglomerative clustering algorithm, they then obtain several clusters with similar questions, answers concepts, askers and answerers, among others. A similar study by Jeon et al. [51] in Naver QA identifies questions that are semantically

[12] https://www.tripadvisor.com/

[13] https://www.lonelyplanet.com/

similar, but very different lexically. They found special clusters of similar queries and URLs using a bipartite graph to learn word translation probabilities.

As we mention in Chapter 1, one of the most important aspects of Twitter is the immediacy of information. We found several studies related to time of questions and answers. For example, Paul et al. [98] observed that roughly 18.7% of Twitter questions receive at least one reply; the first replies come within 5 – 30 minutes and the remainder within the next 10 hours. Liu et al. [72] studied the response rate in the Chinese microblog Sina Weibo and analyzed the characteristics that affect the response rate, such as the number of followers, posting rate, etc., using a mix of QA characteristics and social behavior.

3.3.2 Ranking Task in QA

In this thesis, we train models using Learning to Rank (LTR) methods. We define our problem as a ranking problem because we combine several features in order to find the best ranking. The Ranking task helps to sort the most relevant results (in our case, best answers) based on particular criterion. There are studies of ranking in microblogs, but none of them focus on QA purposes. For that reason, we inspect state-of-the-art studies related to Ranking QA in traditional and microblogs platforms.

Molino et al. [78] extract features and use Learning to Rank (LTR) methods to predict *best answers* in Yahoo! Answers. In this aspect, we find that features used for Yahoo! Answers cannot be directly applied to Twitter, as they are very different platforms. Therefore, we complement this work by studying which features contribute to finding relevant answers in Twitter and if those features are different from those in other platforms. Surdeanau et al. [105] also study non-factoid questions in Yahoo! Answers. They explore using several combinations of features and Natural Language Processing (NLP).

Microblogs. Although, studies on QA platforms address factoid and non-factoid questions, the problem of evaluating non-factoid questions for non-specific cQA platforms is difficult in Twitter. In this thesis we use LTR as a means for learning relevant features for QA using microblogs. In the past, studies have used similar techniques, but for different purposes. For example, Duan et al. [30] rank tweets according to how informative they are by using content features, such as URLs and tweet length.

In general, for evaluating non-factoid questions in Yahoo! Answers previous studies use answers that have been selected by the community as best-answers, as a ground truth. Therefore, their ranking tasks generally consist of predicting the best answer for a question (from the list of answers available for that question) but we do not have this information in Twitter. There are other ranking studies in Twitter, but for different purposes. Yamaguchi et al. [114] propose TURank (Twitter User Rank) to identify authoritative users in Twitter. Through link analysis (user-tweet graph) and interaction features, they find a user's authority score. A similiar approach was made by Weng et al. [111] using an extention of the PageRank algorithm [82] to perform a link analysis. They measure the influence of users in Twitter through the topical similarity between users and analyzing the link structure.

3.3.3 Conversation threads in Microblogs

We focus on conversation threads yielded in the microblog Twitter, which can solve information needs. In particular, we leverage conversation threads in Twitter as a measure of aggregation. To do this, we generate a ranking model based on several features extracted from conversation threads of Twitter. To the best of our knowledge there are currently no studies that specifically address identifying relevant conversations in microblogs to satisfy information needs in Twitter.

In general, a conversation is generated when one (or more) users reply to a tweet. The streaming nature of this platform makes the data volatile and hence a more difficult problem to address. There are related studies about conversations in Twitter; for example, Boyd et al. [14] study message re-posting (or retweets) as if they were conversations, but do not address other types of conversations such as conversation threads. Honey et al. [48] study conversations in Twitter by defining a conversation as mutual user mentions. However, we consider that a mention is not always a conversation. Backstrom et al. [8] study conversations in Facebook and predict user participation in similar conversations.

In general, there are not many studies in this task because the Twitter API only provides tweets and not a conversation thread structure. Hence, it is not easy to build a corpus with this information. In this thesis, we create a corpus and leave it available for future research purposes.

3.4 Other QA Research

Although there are QA specialized sites for asking questions, people often use other media sources. In particular, social networks provide a source of complementary information to search engines [80]. In fact, there are people that prefer to pose their questions in social networks rather than search engines because they tend to trust in their friends and also for the speed of responses [80].

The studies yielded in specialized QA platforms are not always completely reproducible in microblogs. One downside is the length of microblog messages, because answers end up being short in comparison to other QA platforms. Therefore, one must consider other methods to get more complete answers. In addition, users tend to be more specialized on particular topics [42, 50]. In contrast, users in Twitter do not tend to be so specific.

The research community of QA has expanded its focus to other sources where people generate information needs, and not just in traditional QA websites. These studies include similar topics reviewed in section 3.1.

The following section describes the main QA research topics over recent years.

3.4.1 QA Quality

QA quality studies are useful for several purposes, for instance, when inspecting the quality of questions, we can predict whether a question will be answered or not [65]. Also, analyzing the quality of answers allows us to predict either the "best answers" or "best users". Jeon et al. [52] argue that "*estimating the quality of answers is important because some questions have bad answers. It happens because some users make fun of other users by answering nonsense*". In particular, they create a framework to predict quality documents using non-textual features such as the number of answers for given questions, the user activity level[14], answer length, number of questions of the asker, etc. Using a *Kernel Density Estimation* in -*Naver QA*[15], they can distinguish good answers from bad ones, obtaining a good prediction model. Hickl [46] presented a similar study introducing a novel model based on authority and estimating the quality of the "candidate answers" of a given question. In fact, he observed that the question and answer pairs can help in the search for other more precise answers. This model is represented by a database organized as a weighted, directed graph representing the relationship between questions and other questions' answers. Using a statistical learning-based framework, the author can recognize textual entailment[16]. The model is able to predict whether questions will be answered or not.

On the other hand, Li et al. [64] performed a similar study, but focusing on the questions. In particular, they studied how question quality can affect answer quality. They established that: "*high quality questions are supposed to attract great user attention, more answer attempts and receive best answers within a short period*". Using Yahoo! Answers, they analyze several characteristics which could influence in text features and asker profiles. They found that the interaction between the asker and topics are relevant to determine question quality. This means that there are users who ask good questions in one topic, but their question quality in another topic can be poor. Similarly, Agichtein et al. [2] modeled the interaction between questions, answers and users, and found that in general, this kind of interaction is relevant for finding high quality content in social media.

In the same context, Asaduzzaman et al. [6] use question quality to understand why some questions are not answered in Stack Overflow. They observed that the majority of questions are answered in a short time frame, but around 7.5% remain unanswered. The authors argue that understanding the unanswered questions could improve the quality of future questions. Hence, they build a taxonomy and consider user features to train a classifier. Table 3.1 shows the main characteristics about why some questions are not answered: The study also includes a useful model for predicting how long a question could be waiting for an answer, considering question characteristics and user features.

[14] Active level is a rate of asking questions and getting responses from others.

[15] South Korean QA website.

[16] The paper references several works on the method.

Characteristic	Percent
Fails to attract an expert member	22 %
Too short, unclear, vague or hard to follow	17 %
A duplicate question	12 %
Impatient, irregular or inconsiderate members	12 %
Too hard, too specific or too time consuming	9 %

Table 3.1: Asaduzzaman et al. [6] report the main characteristics on why a portion of questions are not answered in Stack Overflow QA platform.

3.4.2 Social Behavior

Social behavior is an important component in QA research. Identifying a pattern of behavior in users can lead to discover interesting research topics. Several studies addressed this aspect to mainly understand the user's motivation for asking questions, getting answers or understanding the topics that people are asking about, etc. For example, Dror et al. [29] realized that some users of Yahoo! Answers provide only one or two answers and then leave the discussion (known as a *churn prediction* problem by the QA community). They create a model to predict whether an answer will be abandoned or not, and employ user features such as the rate activity, gender, personal information, interaction, among others. They also report that users involved in popular contents are more likely to churn the conversation very soon (probably because these types of posts attract more answers). Gyongyi et al [42] studied the characteristics of Yahoo! Answers's users such as activity levels, roles, interest and reputation. They report that people ask *focused questions* triggered by concrete information needs and wait for expert answers. There are also many questions that can generate discussions (as opinions), but it is not common that the user participates in them.

On the other hand, Yang et al. [115] studied the question asking behavior in different cultures. They observed that the culture is an important factor in questions and not only how they are written semantically. Via a particular survey study, they found that Western community (US and UK) have different motivations for answering or asking questions in comparison to Asian countries (China, India). For example, Asians (especially Chinese) tend to ask more questions related to social connections and less rhetorical questions[17]. It means that Asians use mainly QA platforms for professional purposes such as finding jobs or similar opportunities, and less for fun. In contrast, Western countries (as the U.K.) ask more rhetorical questions as well as questions related to fun and entertainment. Furthermore, among 50% - 70% of all countries ask factual knowledge questions (for example, *How can I change my printer cartridge?*).

In addition, we found studies related to sentiment analysis of QA. Kucuktunc et al. [61] inspected user features and the sentiment in Yahoo! Answers. Using the sentiment extraction tool [108], they found correlations of several features with observed attitude and sentimentality. Moreover, they studied sentiment over time and found interesting patterns on weekends, weekdays and also on specific topics. Kentaro et al. [57] explore the utility of sentiment analysis and semantic word classes for improving why-QA (*why* questions) in non-factoid

[17] Questions do not require answers. For example, "*You ruined my life and I'm supposed to like you?*"

questions. Why-QA is a task to retrieve answers from a given text archive for a why-question, such as “Why are tsunamis generated?” The answers are usually text fragments consisting of one or more sentences. They combine these ideas for the purpose of answers. Somasundaran [101] explore the utility of attitude types to enhance QA interaction. By use of automatic classifiers they recognize two main types of attitudes: sentiment and arguing.

We also found approaches that consider the social behavior using connections, the friends list, followers, etc., in Social Networks. For example, Sousa et al. [102] observed that users with large connections separate their connections depending on the topic. Schantl et al. [95] report that user replies to questions in Twitter are influenced by social relations rather than topics.

3.4.3 Expertise and Authority

Active users are typically a small number with an even smaller number of domain experts. Users can have certain expertise o authority on some topics[18]. Detecting expertise or authority helps us find users with special knowledge that could answer similar questions quickly. Pal et al. [83] define QA experts as *“a user who provides a large number of technically correct, complete and reliable answers.”*. In general, experts are more selective than regular users in selecting certain types of questions [83]. They prefer to answer questions which do not already have good answers. It is important to estimate the expertise or authority of users without exclusively relying on user feedback [54].

On the other hand, Bourguessa et al. [13] identify authoritative and non-authoritative users in Yahoo! Answers. They built a weighted directed graph model to represent the askers and the best answerers. The results confirm the effectiveness of the approach identifying authoritative users who are rich sources of knowledge. Jurczyk et al. [55] generate a user’s ranking authority in Yahoo! Answers using the HITS algorithm [76] where the user’s questions are identified as hubs and user’s answers as authorities. They conclude that authoritative users tend to post answers that are popular. A similar study by Jurczyk and Agichtein [54] found authoritative users for specific question categories by analyzing the link structure of the community.

3.4.4 Question Routing

Question routing problems try to route questions to users who can provide the answers. Zhou et al. [124] define the question routing problem as a way to “*determine whether a user will contribute his/her knowledge to answer some question*”. They also address the question routing problem as a classification task. Extracting special characteristics from questions, users, relationships, amount of answers, amount of resolved answers, etc., they develop a variety of local and global features, which are essential in question routing. In the same platform, Li et al. [66] estimate the expertise level of potential answerers and categorize users to define experts per category. In particular, they define category-sensitive

[18] In practical terms, we consider expertise or authority with the same meaning.

language models. Jinwen et al. [41] recommend questions to potential answerers using a model to discover latent topics based on question-answer content. Shtok et al. [99] propose an answerer's recommender for questions in Yahoo! Answers using query-performance prediction and natural language processing. They built user profiles based on the past activity, matching users with new questions. These will be the input of the classifier to predict which users will answer a question. Qu et al. [90], used a Probabilistic Latent Semantic Analysis (PLSA) model to recommend questions to users who are interested in answering them. They analyzed the user's interest of Yahoo! Answers improving the quality of recommendations. They also found good answers as well. Li et al. [65] propose a similar framework to route questions to the best answerers who are most likely to give answers in a short time period. This study combines expertise with availability estimation (time). The final model routes the question to the right answerer considering the performance profiling (history), expertise estimation, answerer ranking and availability estimation.

Regarding social media, we found studies about the routing problem with expertise in Microblogs as well. For example, Liu et al. [73] predict potential answerers to questions in Twitter, based on non-QA features such as answerer's profile and style of posting. In contrast, they report that features such as user characterist ics, popularity and activeness are non-relevant. Souza et al. [103] create a model for query routing on Twitter based on the askers followers and ranks them based on three criteria: knowledge, trust and activity.

Ghosh et al. [38] developed another similar approach and created a tool called "*Cognos*" that infers expertise via crowd wisdom captured by user Lists[19]. Liu et al. [71] created a service called *MoboQ*, which is a location-based real-time social QA. It analyzes Sina Weibo's user questions, location and identifies people who are in the place of the associated question (by temporal analysis), then routes questions to "strangers" that could answer them.

3.5 Other topics

We found related studies of QA but they are not close to this research. We added because these topics are currently being investigated.

Combined topics. We found studies that combine different topics of QA. For example, Zheng et al. [122] evaluate questions and answers in QA Baidu Knows[20] and found that questions with a lot of replies had few good answers (in terms or quality and correctness). The authors propose a framework based on answer quality and also considering the user's interest and expertise. With these two topics, they reduce the waiting time and recommend good answers. Their work resulted in good performance in comparison to other similar techniques.

On the other hand, Fang et al. [33] inspect quality and authority for similar purposes. They find out a user's authority, extracting high quality answers. They differentiate and

[19] Twitter has the option allowing one to follow certain users in personal Lists.

[20] Chinnese QA site.

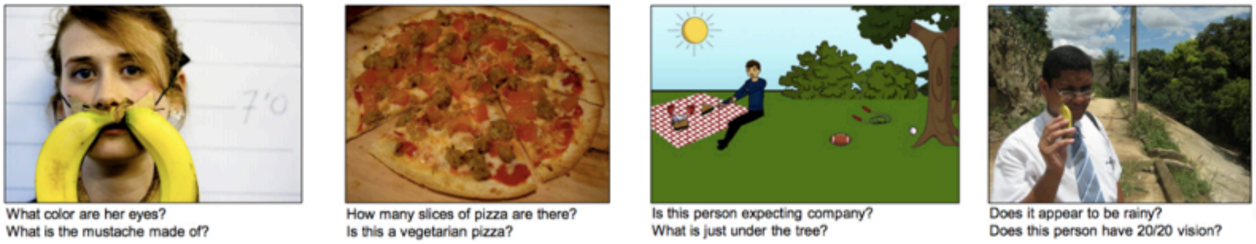

What color are her eyes?
What is the mustache made of?
How many slices of pizza are there?
Is this a vegetarian pizza?
Is this person expecting company?
What is just under the tree?
Does it appear to be rainy?
Does this person have 20/20 vision?

Figure 3.2: Examples of Visual Q&A (VQA). We can ask any questions about the image and the system will answer in natural language. (Images obtained by [5])

Paragraph:
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls **under gravity**. Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".
QA pairs:
- What causes precipitation to fall? **gravity**
- Where do water droplets collide with ice crystals to form precipitation? **within a cloud**

Table 3.2: Example of the SQuAD dataset; a paragraph and QA pairs whose answers are segments of texts (taken by Rajpurkar et al. [93]).

quantify the authority of users for different topics or categories from the content of the questions and the answers. Based on the latter, they can route questions to potential answerers, adding to the user's authority feature.

Visual QA. A new challenge called Visual Question Answering (VQA) has attracted the attention in the Q&A community research. A Visual Question Answering (VQA) system takes as input an image and a natural language question, and the system produces a natural language answer as output. The system needs to understand the image to answer many questions. Figure 3.2 shows examples of VQA taken by Agrawal et al. [5]. More related works were addressed by [5, 117].

SQuAD. The *Stanford Question Answering Dataset* (SQuAD) [92, 93] is a dataset of more than 100,000 questions of Wikipedia articles. The answer to every question is a segment of text. Table 3.2 shows an example of the SQuAD dataset. Some of these features used were; text frequency (TF-IDF), n-grams, parts-of-speech (POS), lexical features and so on. In particular, through a logistic regression, they found that lexicalized and dependency-tree path features were relevant in the performance.

Chapter 4

Exploratory Analysis

Community Question and Answering (QA) sites provide special features for asking questions and receiving answers from users on the Web. Nevertheless, Web users do not restrict themselves to posting their questions exclusively in these platforms. With the massification of online social networks (OSN) such as Twitter, users increasingly share their information needs on these websites.

In this chapter we propose a first approach to extracting relevant information from microblogs for QA purposes. Our findings show that there are special characteristics, features and interactions in microblogs that can be useful for finding answers for new queries posted by users (**RQ1**).

This study was published in the International Workshop on Social Personalisation & Search of the SIGIR 2015 conference [44].

4.1 Context

A rough inspection of QA conversation threads on Twitter yields high redundancy of questions over time, meaning that there is a high chance of finding answers to newly asked questions. This trend of QA usage in Twitter indicates an increasing potential for fulfilling current users' information needs based on similar questions already answered in the past.

None of the previous studies on conversations in Twitter [14, 19, 48] focus on building a knowledge repository of QA in Twitter to answer a question's query. Other researchers have tried to automatically reply to unanswered questions by matching similar questions already answered in the past [87, 99], but they employed the *Yahoo! Answers* platform. The challenge in Twitter is more complex due to the lack of explicit mechanisms to tag good from bad answers as in QA sites.

In this study we address the task of creating a method for retrieving the most relevant historical conversation threads, which can answer a given query q, by leveraging Twitter as

QA repository. In particular, we study how the combination of questions, their replies and Twitter social features (Retweets, Replies, Favorites, etc.) can help determine whether a conversation thread triggered by a question is relevant in terms of information quality, to the particular conversation topic.

For this task, we defined two **research questions**: **(1)** How can we determine whether a conversation thread was resolved (answered) on Twitter? In other words, which factors or features are most relevant to determine that? and, **(2)** Can we recommend relevant conversation threads made in the past to answer a new question? Can we build a ranking model with the most relevant features of **(1)**?

To address this problem, we define the relevance of a conversation thread in terms of its likelihood of providing a complete answer and resolving the information need. Then, to evaluate the importance of each conversation we employed a relevance measure which is based on the feedback provided by the user asking the question on Twitter. Our preliminary findings show that the feedback of the asker plays an important role for evaluating whether a conversation thread had good quality, i.e., whether it was resolved or not. Nevertheless, the noisy nature of tweets makes them complex to analyze, making our problem complex.

To the best of our knowledge, the method proposed in this article is the first attempt to rank conversation threads based on Twitter feedback. Therefore, the **main contributions** of this study are:

1. to propose a methodology for obtaining a set of ranked historical conversation threads that answer a given question, and
2. to identify the main characteristics that influence the quality of a conversation thread.

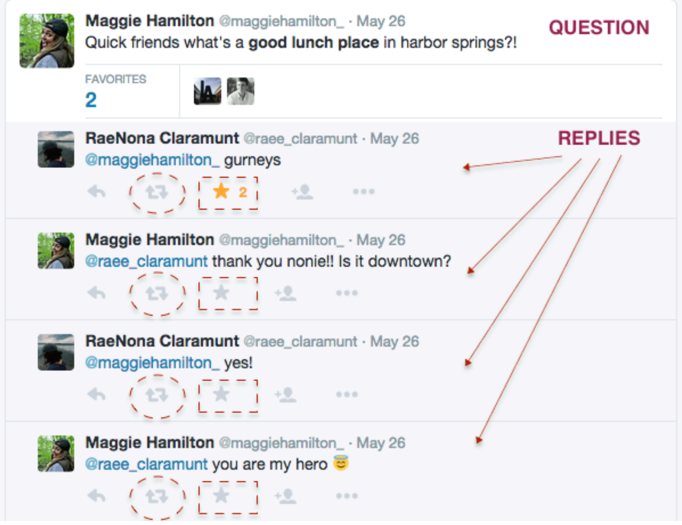
The applications of this work can be useful for any social network with interaction among users to enhance search results. Also, we can use this approach to build a question and answer repository website based on Twitter.

4.2 Ranking QA Threads

In an initial inspection on Twitter, we realized that tweets provide low quality information[1]. For QA purposes, this information is not enough. For that reason, we propose an aggregated *thread-level document model*, which considers **conversation threads** as documents for retrieval. This representation allows us to use aggregated information as documents for answering queries, as opposed to using single tweets. In addition, it can provide more complex information such as different yet complementary opinions for the same question. In short, our hypothesis is that the composition of several tweets into a single thread can provide answers to complex questions. Figure 4.1 shows an example of a QA conversation thread.

In order to identify features that characterize whether a conversation thread has already

[1] Today, Twitter provides 280 characters, though at the time of writing this study (2017), it provided 140 characters.



Maggie Hamilton @maggiehamilton_ · May 26
Quick friends what's a **good lunch place** in harbor springs?!

FAVORITES
2

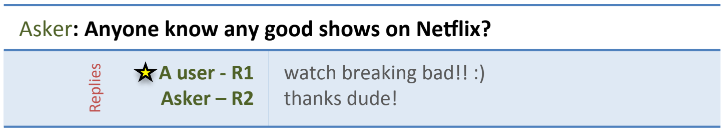
RaeNona Claramunt @raee_claramunt · May 26
@maggiehamilton_ gurneys
★ 2

Maggie Hamilton @maggiehamilton_ · May 26
@raee_claramunt thank you noniel!! Is it downtown?

RaeNona Claramunt @raee_claramunt · May 26
@maggiehamilton_ yes!

Maggie Hamilton @maggiehamilton_ · May 26
@raee_claramunt you are my hero

Figure 4.1: A conversation thread on Twitter formed by one question and four replies (answers). Tweets can be *Re-tweeted* (dotted circles) and *Favorited* (dotted squares).



Asker: **Anyone know any good shows on Netflix?**

Replies		
	☆**A user - R1**	watch breaking bad!! :)
	Asker – R2	thanks dude!

Figure 4.2: An example of a thread with two replies (R1 and R2) where an *Asker* asks a question, a *User* replies, and finally the *Asker* replies back. The star means the *Asker* has marked that reply as Favorite.

resolved an information need, we manually inspected several hundred conversation threads. We define the **relevance of a thread** in terms of *how effective the complete conversation is at answering the information need posed in the initial tweet of the conversation.* This analysis brings us to consider that replies in conversation threads are important at the moment of establishing the relevance of the conversation.

In particular, given Twitter's relevance feedback options, tweets in a conversation thread can be marked by users as *Favorite* and/or *Retweeted*, where the first indicates a special preference and the second indicates that the content has been re-posted by a user. We observe that the feedback provided by the user who posted the tweet, that initiates a thread, called the *asker*, plays an important role indicating the relevance of the thread. Our intuition is that since the asker is very interested in obtaining a good answer to his/her query, a frequent use of Twitter's relevance feedback features will indicate a higher satisfaction. Figure 4.2 shows a simple instance where the asker gives feedback in a thread.

In this case, with an option to determine the level of satisfaction of the asker with a thread, we can evaluate the second reply "*thanks dude!*" of the *asker* using Sentiment Analysis (using NLTK tools[2] we note that the reply has a positive polarity of $67,13\%$). This follows a similar approach by Pelleg et al. [87] for Yahoo! Answers. In the example, if the asker additionally marks the first reply as a *Favorite* (followed by a positive answer) this provides a stronger indication of satisfaction with the reply. We call this type of behavior *positive reinforcement feedback* (PRF), in which the asker indicates their approval for replies to his/her question.

[2] Natural Language Toolkit, `http://text-processing.com`

In an initial inspection of our dataset we have identified at least 5 other similar types of PRF which are recurrent over time in QA threads.

4.3 Problem statement

We can define our problem as: given a new question q^* we retrieve a set of its top-k similar conversation threads. We do so initially by retrieving threads with the highest cosine similarity of their initial tweet q^*. We denote this set of similar threads as $T = \{th_1, \ldots, th_k\}$. Each thread is given by $th_j = (q_j, R_j)$, in which q_j is the initial tweet or question of th_j and R_j is the set of replies received for q_j. Then, for each thread th_j we compute its absolute relevance $rel(th_j)$ that indicates the level of satisfaction of the asker of q_j with the overall replies received in th_j. Initially we estimate $rel(th_j)$ as:

$$rel(th_j) = \textit{count of positive reinforcement instances in}\, th_j.$$

Using the value of $rel(th_j)$ we re-order the set T obtained for q^*. We only take the top-k elements with highest $rel(th_j)$. We do not use a threshold value because each thread presents different level of feedback, hence we cannot define a fixed value.

4.4 Methodology

We retrieved questions directly from Twitter. Then, for each question $q*$, we retrieved similar threads two weeks later. Since our goal is to have sets of similar questions in the dataset and question threads are very sparse in the public stream, we conduct a focalized crawl for threads. More specifically, we retrieve question threads for our dataset using the following iterative process: 1) we search in Twitter for a list of common words used in questions, 2) filter all of the results (tweets) that corresponded to a question q^*, 3) for each q^* perform an additional search to retrieve similar questions-threads th_j. The full process (1)-(3) was conducted between March 31, 2015 and April 27, 2015.

4.5 Experimental Setup

For search and filter steps we used the Streaming API[3] using a traditional rule-based approach [32, 80, 121]. We also defined additional rules of questions that we need, because not just any question is useful in our task. We collect questions that require answers (related to information needs or factual knowledge), questions that are not affected by time, recommendation questions, suggestion requests, questions in English, and opinion questions. For instance, in our dataset we kept questions such as: "*does anyone know cheap places to stay in London?*" or "*can anybody recommend me good restaurants in Santa Monica?*".On the other

[3] The streaming API captures 1% of the Twitter volume in real time.

hand, we discard questions such as: "*anyone got an iPhone 5 for sale?*" or "*Anyone know what time the Mayweather fight starts*??". The first is not a factual knowledge question and the second is affected by time.

Regarding the similar question-threads retrieval, we built the set of similar past threads th_j of q^*. Since Twitter API restricts obtaining the complete thread, we must first retrieve similar tweets and then the replies, if they exist. We have retrieved all the q_j that are similar to q^* from the **Search API**[4] (the retrieval is based on the main keywords of q^*). Then, we retrieved the replies R_j of q_j to build the thread structure. We adapted a development made by Crepaz[5] that can get replies through the Twitter mobile webpage. Finally, we calculate the relevance $rel(th_j)$ for each thread based on the PRF.

In order to show evidence of the usefulness of our proposal, we collected a **dataset** of tweets in the English language. This preliminary Twitter dataset contains 721 questions taken by Twitter (q^*) and $152,721$ conversation threads (th_j) related to q^* (conversation threads could be tweets as well). Actually, we retrieved more than 1,000 questions from the API using the question mark "?". However, we selected just questions that meet the above restrictions (i.e.:not affected by time, not recommendation questions, etc.). The questions were collected from March 31, 2015 to April 10, 2015.

4.6 Results

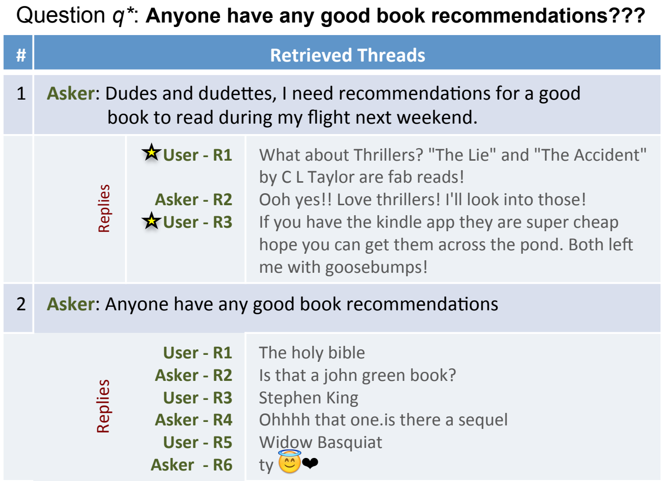
By both automatic analysis and manual inspection of our dataset, we identified common patterns of QA conversation threads. We describe three of the most recurrent examples and how our ranking methodology works in each case.

Case 1. Figure 4.3 shows the top 2 similar threads retrieved by our approach sorted by relevance (high to low), given the initial question q^*: "*Anyone have any good book recommendations???*" (taken literally from Twitter). The first thread contains three replies and two of them were marked as Favorites by the asker (see the starts). Notice that the reply R1 (of the thread #1) was marked as Favorite by the asker and followed by the asker (reply R2) with positive sentiment (*"Ohhh yes!! Love thrillers! I'll look into those"*). The sentiment analysis of R2 has 76% of probability that the text presents positive polarity. That means that the thread presents PRF. The reply R3 of the thread #1 was marked as favorite by the asker, but it is not followed by any tweet of the asker. On the other hand, the thread #2, presents a positive sentiment in the reply R6[6]. Although the asker uses feedback elements (positive expression in the reply R6), the thread #2 does not present the structure to be PRF. Hence, the relevance is lower than the first thread.

[4] The search API retrieves tweets posted within a week of the time the query was issued.

[5] http://adriancrepaz.com/twitter_conversations_api (not longer available).

[6] "ty" means "thank you".

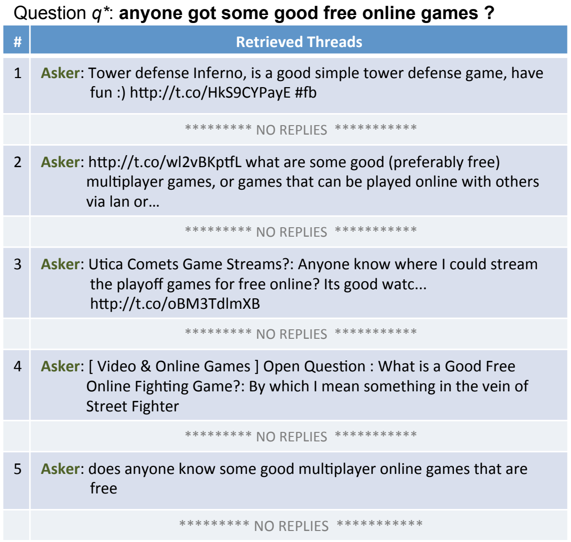


Question *q**: **Anyone have any good book recommendations???**

#	Retrieved Threads		
1	**Asker**: Dudes and dudettes, I need recommendations for a good book to read during my flight next weekend.		
	Replies	☆User - R1	What about Thrillers? "The Lie" and "The Accident" by C L Taylor are fab reads!
		Asker - R2	Ooh yes!! Love thrillers! I'll look into those!
		☆User - R3	If you have the kindle app they are super cheap hope you can get them across the pond. Both left me with goosebumps!
2	**Asker**: Anyone have any good book recommendations		
	Replies	User - R1	The holy bible
		Asker - R2	Is that a john green book?
		User - R3	Stephen King
		Asker - R4	Ohhhh that one.is there a sequel
		User - R5	Widow Basquiat
		Asker - R6	ty 😇❤

Figure 4.3: Case 1. Given a question q^*, we show the top-2 relevant threads. The stars mean that the tweet was marked as Favorite by the Asker.

Case 2. Figure 4.4 presents another recurrent case. Given an initial question q^*, the threads retrieved are just the initial tweet q_i, without replies. But, on further observation, the retrieved tweets still can answer the question q^*. When this occurs, we sort the tweets depending on whether they contain URLs (tweet relevance should be high when it contains a URL [20]). In our dataset, the amount of threads without replies are 69.9%. We notice that after the third tweet the tweets do not clearly reply to the initial question $q*$.

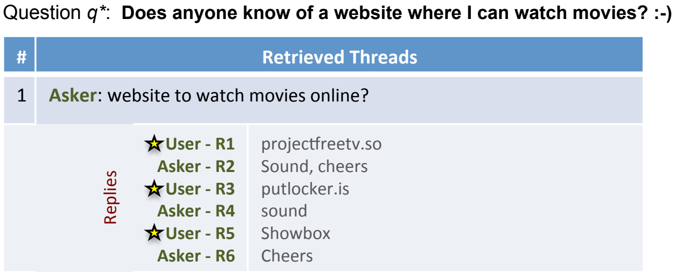


Question *q**: **anyone got some good free online games ?**

#	Retrieved Threads
1	**Asker**: Tower defense Inferno, is a good simple tower defense game, have fun :) http://t.co/HkS9CYPayE #fb
	********* NO REPLIES ***********
2	**Asker**: http://t.co/wl2vBKptfL what are some good (preferably free) multiplayer games, or games that can be played online with others via lan or...
	********* NO REPLIES ***********
3	**Asker**: Utica Comets Game Streams?: Anyone know where I could stream the playoff games for free online? Its good watc... http://t.co/oBM3TdlmXB
	********* NO REPLIES ***********
4	**Asker**: [Video & Online Games] Open Question : What is a Good Free Online Fighting Game?: By which I mean something in the vein of Street Fighter
	********* NO REPLIES ***********
5	**Asker**: does anyone know some good multiplayer online games that are free
	********* NO REPLIES ***********

Figure 4.4: Case 2. The top-5 threads retrieved are just tweets (without replies), but we can sort them by URLs.

Case 3. Conversation threads can have high relevance if they had more instances of PRF within the same thread. Figure 4.5 shows this case, where the thread presents PRF twice in one thread. The reply R1 has been marked as Favorite by the Asker. The reply R2 was made by the Asker with 60% of positive sentiment. Hence, the replies R1-R2 present PRF. The replies R3-R4 also present PRF (R3 was marked as Favorite by the Asker and reply R4 returned 54% of positive sentiment). Although reply R5 has a Favorite made by the Asker, the reply R6 has a neutral sentiment. Therefore, they do not present PRF.



Question *q**: **Does anyone know of a website where I can watch movies? :-)**

#	Retrieved Threads	
1	**Asker**: website to watch movies online?	
Replies	★**User - R1**	projectfreetv.so
	Asker - R2	Sound, cheers
	★**User - R3**	putlocker.is
	Asker - R4	sound
	★**User - R5**	Showbox
	Asker - R6	Cheers

Figure 4.5: Case 3. Two types of *positive reinforcement feedback* (PRF) in one thread: R1 was marked as *Favorite* by the asker and R2 has positive sentiment. The same is observed with replies R3-R4. R5 has a favorite but R6 presents a neutral sentiment.

4.7 Final comments

This first approach shows that there is relevant information on microblogs for QA purposes. We proposed a method to retrieve and rank historical conversations threads to answer recent questions. In addition, we use an aggregated thread-level document model, which considers conversation threads as documents for retrieval. This representation allows us to use aggregated information as documents for answering queries, as opposed to using a individual tweet.

We found some features that are relevant to find answers in conversation threads such as social signs. However, we discard the sentiment feature for a future analysis due the dynamic of the text in tweets (misspelled words, short text, among others.). We also discard asker feedback interactions due the API restrictions of Twitter.

In the next chapter we inspect additional features of microblogs related to content and social interaction, and also we study their influence for determining relevant answers.

Chapter 5

A model for Twitter-based QA

We demonstrated that it is possible to use microblogs as QA retrieval resource. In this chapter we build a framework which uses sets of features to rank relevant threads. In particular, we propose a way to study which microblog features have the most influence for determining relevant answers. To conduct this study, we model our research problem as a ranking problem, in which the main goal is to rank relevant answers in the top result positions. We also describe the steps to collect the information, extract features, train a ranking model and evaluate them using ranking metrics.

Formalization. We define our problem as follows: let q^* be a question corresponding to an information need formulated by a user. Let $Q^* = \{q_1, q_2, \ldots, q_n\}$ be the set of possible query formulations of q^*. We define query formulations in Q^* as any variation of the initial input query q^* that allows us to retrieve a set of threads (i.e., documents) that are candidate answers for q^* [53]. Then, for each $q_\mathrm{i} \in Q^*$, we extract all of the threads (documents) that match q_i content in a given microblog dataset. In particular, we say that a thread t_i matches query q_i when t_i contains all of the terms in q_i. Next, let $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ be the set that contains the union of the sets of threads that match the query formulations in Q^*, and therefore by extension, which match q^*. Hence, the goal of our research is, for a given question q^*, to rank the threads in $\mathcal{T}$ according to their relevance for answering q^*.

5.1 Methodology

To accomplish these tasks, we propose a methodology that includes a retrieval model, feature extraction and a ranking process. Our main goal is to demonstrate that Twitter stores valuable information that can satisfy information needs related to QA. To address this task, we divide the study in two parts; first, we create a ranking model using well-formed questions, later, extending the model for more complex questions.

In the **first part**, our intention is to determine the most relevant features to identify candidate answers. We will study these features and combine them in different forms to create a ranking model. For this step we evaluate the performance of our model using **Factoid questions** (factoid QA) which are questions with one answer (a fact). For example, "*Who painted the Mona Lisa?*" (R: Leonardo da Vinci) or "*What country made the statue of liberty?*" (R: France). In particular, we propose to use a well known dataset of factoid questions. We extract keywords from questions (query formulations) and retrieve candidate answers from Twitter. These candidate answers are indeed threads because they provide more information for QA purposes (as we describe in Chapter 4) than single tweets. Hence, from these threads we extracted social, time and content features, among others. We then analyze the importance of each one and the combination of them for ranking purposes. We then conduct an automatic evaluation of the model based on our ground truth and the relevance of threads. Finally, we train a ranking model using a Learning to Rank framework [70].

In the **second part**, we study more complex questions that require more than one answer: **non-factoid questions** (non-factoid QA). They are questions that require answers with opinions or experiences, hence they are subjective. For example, "*Which games are good for ps4?*" or "*Anyone have a good Netflix series recommendation?*". To address these tasks, we focus on recommendation questions. We chose such questions as they are among the most asked in Twitter [80]. Figure 1.1 shows two examples of these types of questions and how we can present answers in Twitter. We follow the same pipeline of the first part, but unlike factoid QA, we consider real non-factoid recommendation questions taken directly from Twitter extracting only relevant features (studied in the first part as well). We next retrieve candidate answers using the same query formulations and apply the best model learned in the first part. We will evaluate the behavior of the model over more complex questions using well-known ranking metrics. In particular, we evaluate if it is possible to apply this model to other kinds of non-factoid questions using transfer learning [84, 85, 123].

In summary, the first part provides a quantitative evaluation based on standard QA ground truth datasets, and the second one is an exploratory analysis based on a manual evaluation. Our intention is to eventually use the factoid QA task as a proxy to our goal of answering more complex questions (non-factoid).

5.2 QA Ranking Model Pipeline

We defined a sequence of steps to build this ranking model framework. As we mentioned in the methodology, we built the model using Factoid questions. Jurafsky and Martin [53] defined as *factoid* questions those that require one answer: "We call the task factoid question answering if the information is a simple fact, and particularly if this fact has to do with a named entity like a person, organization, or location.". Examples of factoid questions are: "*Who was the first American in space?*", "*Where did Dylan Thomas die?*", or "*What is the capital of California?*". In addition, factoid questions are usually short, much like tweets.

Using different sets of features with LTR methods, we build several models that rank tweets and conversation threads as potential answers to a set of given factoid questions. A good model ranks relevant tweets or threads to the top of a list, and we define a tweet or thread as relevant of it contains the correct answer to a given question.

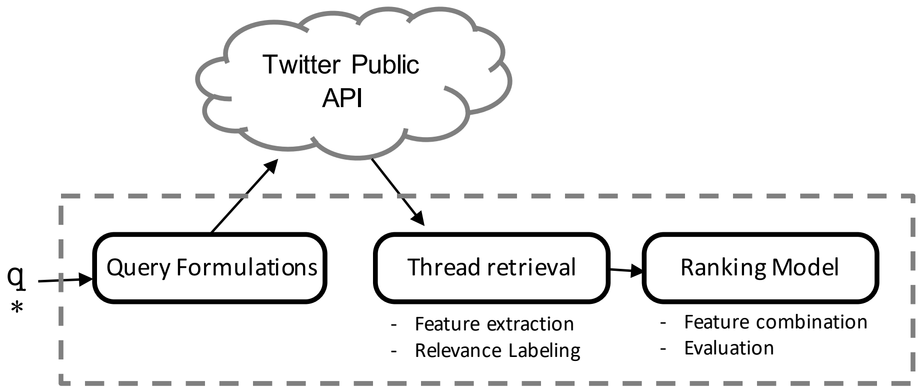


Figure 5.1: A pipeline to train a model using Learning to Rank (LTR). Given a question q^*, we retrieve and build threads using query formulations and Twitter API. We label each thread as relevant or non-relevant using a dataset of QA as ground truth. Finally, we train a model and evaluate the rankings.

Figure 5.1 shows three main steps that we follow to build the ranking model framework. We describe each one as follows:

5.2.1 Query Formulations

We extract specific keywords from questions in the ground truth. These are useful for the search process to increase the recall of similar threads on Twitter.

We define four query formulations that offer four ways of retrieving information in Twitter. For each question q^* we apply the following formulations:

1. $\mathbf{QF_1}$**:** Corresponds to the original question as it was formulated by the user q^*, without any changes or filters.
2. $\mathbf{QF_2}$**:** Corresponds to q^* after lowercase and whitespace normalization, removal of non-alphanumerical characters and terms with only one character.
3. $\mathbf{QF_3}$**:** Corresponds to QF_2 after the additional removal of stopwords, with the exception of terms in the **6W1H**[1]. For example, the question q^* = "What is the scientific name of tobacco?" becomes QF_3 = "what scientific name tobacco".
4. $\mathbf{QF_4}$**:** Corresponds to q_3 without the **6WH1**. In the previous example, q^* would be transformed to q_4 ="scientific name tobacco".

We apply lowercase normalization and removal of non-alphanumerical characters in q_2 and q_3. Finally, the retrieved tweets are $\{q_1 \cup q_2 \cup q_3 \cup q_4\}$.

We acknowledge that there are other important tasks besides ranking in QA retrieval such as, creating the best possible query formulations [15] and selecting the passages within a text that contains the answer to a question [53]. However, at this moment we consider those problems as beyond the current scope of our work.

[1] **6W1H** corresponds to 5WH1 with the addition of the term "Which" (i.e., Who, What, Where, When, Why, Which and How).

5.2.2 Thread Retrieval.

In an ideal scenario, our answers would come from a large historical Twitter dataset, or from the complete data-stream. However these types of data repositories are not available to us at this time, hence we approximated them by applying our method to the data retrieved using the Twitter Search API [2] as an endpoint, which provides access to a sample of the actual data.

As shown in Figure 5.1, thread retrieval is performed sending query formulations to the Twitter API. In particular, we retrieve threads following the same idea of Chapter 4 and Herrera et al. [44]. For the retrieval process we follow the following procedure; for each question q^* we retrieved from Twitter its related tweets using our query formulations in Q^*. If the retrieved tweet is part of a conversation thread we also retrieve its full thread, if not, we keep with the single tweet. The complete set of Twitter threads obtained using Q^* corresponds to $\mathcal{T}$ the set of candidate answers for q^*.

Feature Extraction. Once we have the formed threads, we extract several features based on content and social signs. We performed a review of features used in prior work on traditional QA and adapted those that could be applied to microblog data (such as [10, 30, 78]). In addition, we inspect approximately $1,000$ QA conversation threads yielded in Twitter (i.e., threads in which their initial tweet is a question) in order to qualitatively estimate which features could potentially be influential in determining if the initial question was answered within the conversation thread itself [44]. This inspection allowed us to identify features that could potentially help determine whether the question in the QA thread was answered inside the thread itself or not. Our methodology uses any type of thread to gain insight into QA behavior. Including threads whose initial tweet is paraphrased as question and also threads related to a discussion (without an initial question).

Relevance Labeling. As described above, we define an answer as **relevant** if it contains a correct answer to the original question. Along these lines, the ground truth must provide the question and the correct answer.

5.2.3 Ranking Model

We train our model using LTR methods. We test several combinations of features and use the thread relevance as the class/label of each instance.

Feature Combinations. We use the heuristic called **Forward Feature Construction** [100] in order to find the best feature combination. It is mainly used for dimensionality reduction purposes. The idea is to start with one feature and progressively add another each time. We keep the feature (or the set of features) that produces the highest increase in performance.

[2] `https://developer.twitter.com/`

We adapt this heuristic to our problem and we define it as follows: let f_{i} be a feature set (e.g. part-of-speech features), $F = \{f_1, f_2, \dots f_n\}$ the set that contains all of our feature sets, and PBC (initially, $PBC = \emptyset$) the partial best feature set combination:

1. We run the factoid task evaluation for each feature set in F using each LTR model.
2. We choose the feature set f_{i}^* which produces the best MRR@10 and add it to the set PBC (i.e., $PBC = PBC \cup f_{\mathrm{i}}^*$) and remove f_{i}^* from F (i.e. $F = F - f_{\mathrm{i}}^*$).
3. We again run the same evaluation using the resulting PBC in combination with each remaining feature in F (i.e., $(PBC \cup f_1), (PBC \cup f_2) \dots (PBC \cup f_n)$).
4. We repeat the process from step (2) until there is no a improvement in the MRR@10 value.

Through this heuristic it is possible to show the contribution of each feature and then to build the best combination.

Evaluation. We evaluate the model using well known ranking metrics defined in Chapter 2.2.3. We mainly use the MRR metric, which evaluates the ranking based on the position of the first relevant thread in a list.

Summarizing, we have described the process to generate a model that shows the best selection and combination of features. In the next chapter, we perform an experiment using well-known datasets of questions and answers. We then retrieve relevant answers from Twitter following the proposed pipeline.

Chapter 6

Factoid QA Task

In this Chapter, we conduct an experimental validation using well-known ground truth datasets of QA. We study the discriminating features that identify relevant answers to a particular query.

To conduct this study, we follow the model (pipeline) described in the previous chapter and set up a learning to rank framework using specific types of questions (factoid QA). Our results show that we are able to identify the importance of different features and their combinations (**RQ2**). Using the MRR metric we improve our baseline by more than 62%.

The results of this chapter were published in the conference ECIR 2018 [45].

6.1 Ground Truth

We inspected several QA datasets in order to train our model. We built the ground truth QA dataset for our experiments based on several public datasets by the TREC[1] QA challenges[2] and factoid-curated repository of questions for benchmarking Question Answering systems[3].

For TREC datasets, we used four QA datasets which contain factoid questions with their respective answers: *TREC-8 (1999)*, *TREC-9 (2000)*, *TREC-2004* and *TREC-2005*. Three of these datasets, TREC-9, TREC-2004 and TREC-2005, provide factoid questions, the answers and regular expressions of answers. TREC-8 provides the questions and the answers, but the regular expression of answers is missing, hence we constructed them manually.

The factoid-curated dataset provides the factoid questions, the answers and the regular expressions of answers. The regular expressions of answers are very important in our dataset allowing us to address an automatic evaluation in order to match correct answers in text.

[1] TREC: Text Retrieval Conference

[2] `http://trec.nist.gov/data/qamain.html`

[3] `https://github.com/brmson/dataset-factoid-curated`

In summary, we collected $1,634$ factoid questions joining TREC datasets and curated-dataset. From this set, we manually eliminated questions (with their answers), which present the following characteristics:

- Time-sensitive questions, i.e., questions with answers that can change over time. For example, "*What is the population of the Bahamas?*".
- Imprecise questions: for example, "*what is the size of Argentina?*".
- Not phrased as a question: for example, "define thalassemia.".
- Questions that require a list of answers: for example, "*What countries have tsunamis struck?*".
- Questions referred to other questions: for example, "*What books did she write?*" (in reference to another previous question).
- Questions whose length was over 140 characters[4].

Applying all the previous filtering, we considered $1,051$ questions[5].

6.2 Thread Retrieval

We prepared the data for the training process and hence, followed the steps defined in the pipeline shown in Chapter 5.

From the Twitter API, we built the threads using our query formulations for each question of the ground truth. Overall, we found candidate answers for 491 of the questions (i.e., 47% of $1,051$). The remaining questions had no candidate answers, or none of the candidate answers matched the correct answer. Examples of such questions are:

- *"What was the ball game of ancient Mayans called?"*
- *"How many plays were there in Super Bowl XXXIV?"*

For these 491 questions we were able to retrieve a set of candidate answers $\mathcal{T}$ for which at least one of the threads (or tweets) contained the correct answer for q^*. Examples of those questions are:

- *"What city is Purdue University in?"*
- *"When was Queen Victoria born?"*

We noted that our initial factoid dataset does not contain current topics, which are much more likely to be discussed in social media (the most recent TREC dataset is from 2005, and the oldest tweet we could retrieve is from 2007). This could explain the small overlap between TREC questions and Twitter, which is characterized by discussing more timely subjects [62].

[4] We are aware that Twitter recently increased the amount of characters to 280, but through we carried out this research before this announcement. However, this change should not affect the main results of this work.

[5] The dataset is available in `https://github.com/jotixh/ConversationThreadsTwitter`

Regarding the **relevance labeling**, we used the regular expression provided in our ground truth. The mayor advantage is that we automatically can inspect if the regular expression of the correct answer is inside the threads or not. If it is, we label the thread as **relevant**, otherwise, **non-relevant** . Hence, the goal is to rank relevant answers (i.e. correct answers) first, and study which microblog features have a stronger effect on good rankings.

In order to improve and balance the dataset for the learning process, we removed low threads that present all of these three conditions:

1. if it is non-relevant ,
2. if the thread has no replies (just tweets) and,
3. if the cosine distance between the query and the thread is $\leq$ 0.3.

Finally, applying this filter, the resulting dataset contained $33,873$ conversation threads with $63,646$ tweets. Table 6.1 shows a description of the final dataset.

Number of TREC Questions	491
Questions with $\geq$ 50 candidate threads	49
Questions with $\geq$ 10 candidate threads	146
Questions with $<$ 5 candidate threads	277
Number of threads	33,873
Number of tweets	63,646
% of tweets that are part of a thread	46.7%
Avg. replies per thread	0.9
Number of users involved	38,453
Number of relevant threads	9,406
Number of non-relevant threads	24,467

Table 6.1: Ground truth dataset description.

6.3 Feature Extraction

We performed a review of features used in prior work on traditional QA and adapted those that could be applied to microblog data. We used state of the art features and propose others.

This study includes a **word embedding** representation of questions and threads. Several works in NLP [22, 58] have significantly improved their performance using word-vector embeddings such as Word2Vec [77] rather than the traditional vector space model with TF-IDF weights. In particular, we used Word2Vec [77] with a pre-trained model on 400 million tweets provided by the author in [40]. Each vector was composed of 300-dimensions and since the documents in this model are matrices rather than vectors, we used the max-over-time pooling

Feature ID	Description
D_TFIDF_N (ngram $N = \{1, 2, 3\}$)	Cosine, Manhattan, Euclidean and Jaccard distance between q^* and the thread t_i.
D_WEMB	Cosine, Manhattan, Euclidean distances between q^* and the thread using Word2Vec.
SOCIAL	
SOCIAL_N_REPLIES	Number of replies.
SOCIAL_NDIF_REPLIERS	Number of different repliers.
SOCIAL_RATE_FAVORITES	Average of favorites.
SOCIAL_RATE_RETWEETS	Average of retweets.
SOCIAL_MENTIONS	Number of mentions.
SOCIAL_NDIF_MENTIONS	Number of different @mentions.
SOCIAL_NDIF_HASHTAGS	Number of different #hashtags.
USERS	
USERS_NDIF_FOLLOWERS	Number of followers of users of the thread (unique).
USERS_NDIF_FOLLOWINGS	Number of followings of users of the thread (unique).
USERS_AVE_AGE	Age average between replies and user date of Twitter.
USERS_RATE_VERIFIED_ACCOUNT	Average number of users with verified account.
CONTENT	
CONTENT_NDIF_URLS	Number of different URLS.
CONTENT_N_WORDS	Number of words.
CONTENT_DENSITY	Number of words / number of tweets.
CONTENT_RATE_UPPER	Average number of uppercase in threads.
CONTENT_RATE_LOWER	Average number of lowercase in threads.
CONTENT_EMOTICONS_POS	Number of positive emoticons.
CONTENT_EMOTICONS_NEG	Number of negative emoticons.
CONTENT_EMOTICONS_NEU	Number of neutral emoticons.
CONTENT_RATE_VOCAB	Rate of well written words.
TIME	
TIME_LIFESPAN	Time difference between the first tweet and the last.
TIME_AVERAGE	time average between each tweet of the thread.
POS	A set of part-of-speech tags.
REPW	Rate of the words that are in the 50% more representative words.
WEMB_Q	Explicit vector representation of the question q^* using word2vec (300 dim.).
WEMB_THR	Explicit vector representation of the thread using word2vec (300 dim.).

Table 6.2: Feature sets used for the ranking task. In some cases, they are a set of features and in other cases they are single. For word embeddings (WEMB), we use a pre-trained model of 400 million tweets by [40].

introduced by [22] to flatten our document-matrices to document-vectors. We tested with other models such as a pre-trained model of Google News [6] and word vectors trained on Wikipedia using fastText[7], but after preliminary experiments, the Twitter pre-trained model performs better than others. In short, we used a pre-trained model to represent threads and questions as vectors, and also, to calculate distance metrics.

Table 6.2 summarizes the features used in our study, grouped by type. Note that features are computed at a thread level. We describe them in details as follows:

- **Distance features (D_TFIDF_N and D_WEMB)**: These features are based on four well known distance metrics between a thread t_i and a query q^*.

[6] https://code.google.com/archive/p/word2vec/

[7] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

- Cosine distance:

$$Cosine(\vec{q^*}, \vec{t_\text{i}}) = 1 - \frac{\vec{q^*} \cdot \vec{t_\text{i}}}{\|\vec{q^*}\| \, \|\vec{t_\text{i}}\|}$$

- Manhattan distance:

$$Manhattan(\vec{q^*}, \vec{t_\text{i}}) = |\vec{q^*} - \vec{t_\text{i}}|$$

- Euclidean distance:

$$Euclidean(\vec{q^*}, \vec{t_\text{i}}) = \|\vec{q^*} - \vec{t_\text{i}}\|$$

- Jaccard similarity:

$$Jaccard(q^*, t_\text{i}) = \frac{q^* \cap t_\text{i}}{q^* \cup t_\text{i}}$$

These features are computed using TFIDF and word2vec representations between the query $\vec{q^*}$ and each thread $\vec{t_\text{i}}$. For TFIDF, we denote the computation of these four metrics as **D_TFIDF_**N where N is the n-gram used; $N = \{1, 2, 3\}$. For word2vec, we carried out the same computation, but excluding Jaccard. We denote it as **D_WEMB**.

- **Social-based features (SOCIAL)**: These features are based on the social interactions observed in a conversation thread (i.e., thread-level features). These include: number of replies in a thread, number of different users that participate, fraction of tweets with favorites/retweets/hashtags, number of user mentions, and number of different user mentions.

- **User-based (USER)**: This feature set considers properties of the users that participate in the same conversation thread. These include: total number of followers and followees of the users that participate in a thread, the fraction of users in the thread that have a verified account, the average *age* of the users in a thread. This latter feature is the difference between the date of creation of the Twitter user account and the date when the tweet was posted. We adapt these features from [75, 88].

  The formula is as follows; let $T_w = \{t_1, t_2, \ldots, t_n\}$ be a thread with n tweets, $date(t_\text{i})$ is the date when the tweet t_i was published and $UserDate(t_\text{i})$ is the date when the user of the tweet t_i creates an account in Twitter. The user average age (UAA) of the thread is defined by:

$$UAA(T_w) = \frac{1}{n} \sum_{t_1}^{t_n} \Big[date(t_\text{i}) - UserDate(t_\text{i}) \Big]_{days}$$

  The idea to use the number of followers has been adapted from [30].

- **Content-based features (CONTENT)**: This set of features refers to content properties of a thread. These include: the number of different URLs in the thread, the number of words (removing URLs and punctuation), the length of the thread $\frac{\#\ words}{\#\ tweets}$ (considering only words with size ≥ 1), the fraction of uppercase and lowercase letters, the number of positive/negative/neutral emoticons, and the average number of words in English[8].

  These features have been adapted from those proposed by [10, 30, 75, 78, 88, 97]

[8] We use the English word corpus of NLTK (http://www.nltk.org/).

- **Part-of-speech features (POS)**: These features are based on parts of speech tagging. We compute the frequency of each high-confidence POS tag in a conversation thread, using the Twitter-specific tagger *TweetNLP* by [81]. For example, the tweet: "*Arya, Sansa and Jon need to reunite. #GameofThrones*" has ten tags, where: "Arya Sansa Jon" are proper nouns, "." and "," are punctuation, "and" is a conjunction, "to" is a preposition, "need reunite" are verbs, and "#GameofThrones" is a hashtag. *TweetNLP* has 25 categories for tagging words. We annotate the tag frequency of each thread with a confidence value greater or equal 0.7.

  The complete list of tagsets is in [81]. However, some of these tags are related to social features: #hashtag, @mention, URLs and emoticons. We decided to remove these tags because we use them in social features. Therefore, we use this set of features with 21 tags.

- **Representative words (REPW)**: This feature corresponds to the fraction of *representative words* that are contained in a thread. *Representative words* are words contained in the top-50% most frequent terms over all threads in the training data (excluding stopwords). We evaluated other variations for this feature (without removing stopwords, using Term Frequency, etc.) and selected the best one.
- **Word embedding representation (WEMB_Q and WEMB_THR)** : We use the explicit vector representation of the query and threads using Word2Vec. As mentioned previously, we use a pre-trained dataset of tweets with 300-dimension vectors. As mentioned, we computed the vector representation using max-over-time pooling to flatten our document-matrices to document-vectors.
- **Time-based features (TIME):** These features include time-based characteristics of the thread, such as: time-lapse between the first tweet in the thread and the last, and the average time between tweets in a thread.

6.4 Experimental Setup

We built several models that rank tweets and conversation threads as potential answers to a set of given factoid questions.

We used the LTR software library Ranklib of the LEMUR project[9] for this task and ran the experiments using the default *Ranklib* parameters for the four methods detailed in Chapter 2.2.3.

The parameters used in LTR methods are as follows:

- **Ranknet** : 100 epochs, 1 hidden layer, 10 hidden nodes per layer and a learning rate 5.0×10^{-5};
- **MART**: $1,000$ trees, 10 leaves and a learning rate of 0.1;
- **RankBoost**: 300 rounds and 10 thresholds candidates; and
- **LambdaMart**: $1,000$ trees, 10 leaves and 256 threshold candidates.

[9] `https://sourceforge.net/p/lemur/wiki/RankLib/`

6.4.1 Evaluation methodology

To reduce the probability of obtaining significant differences among the LTR methods only by chance, we relied on bootstrapping [31]. Rather than having a single train/test split of the dataset, we sampled with replacement 30 random collections. Each collection was next divided into 70% of the questions for training (with their respective tweets/threads) and 30% for testing. We also evaluated different combinations of sets of features in every experiment. For each combination we computed MRR@10 and $nDCG$@10 (defined in Chapter 2.2.2). It means, we evaluate just the top 10 candidate threads. In each case we report the mean value over the 30 bootstrapped collections.

In addition, we analyze the impact of each set of features in the model. For this purpose, we use the metric *Empirical Feature Efficiency* (EFE) by Sanchez [94]. It compares the contribution of each set of features regarding the best combination.

We adapted this metric to our problem as follows:

$$EFE = \frac{MRR(M^{best}) - MRR(M^{\mathrm{i}})}{MRR(M^{best})} \tag{6.1}$$

Where M^{best} is the MRR@10 of the best combination features and M^{i} is the MRR@10 of each set of features defined in Figure 6.2.

The metric is expressed in terms of the percentage of the best combination.

6.4.2 Baselines

We compared our approach to the following methods:

- **Twitter Search:** This method lists results from Twitter's search interface. Results are obtained by searching for each query in Q^* using the "latest" option, which lists messages from the most recent to the oldest message. The results obtained for each query are then joined in chronological order. However, this method is not reproducible since it works like a "black box" from our perspective.
- **REPW:** This method uses the feature REPW, described previously, with the best performing LTR model from our experiments.

  Experimentally, this method behaves as an upper bound of the "Twitter search" method with the advantage that it can be reproduced.
- **BM25:** The Okapi weighting BM25 is widely used for ranking and searching tasks [76]. We use the BM25 document score in relation to a query.

  The computation of BM25 is as follows: given a query Q with terms $q_1, q_2, ..., q_n$, the BM25(d) score of a document d (in our case, threads) is computed as:

$$\sum_{\mathrm{i}=1}^{n} IDF(q_{\mathrm{i}}) \frac{f(q_{\mathrm{i}}) * (k_1 + 1)}{f(q_{\mathrm{i}}) + k_1 * (1 - b + b * |D|/\overline{D_{ave}})}$$

Where $f(q_i)$ is the number of times that term q_i occurs in document d, D is the length of the document D (number of words) and $\overline{D_{ave}}$ is the average number of words per document. The b and k_1 are free parameters for Okapi BM25. In particular, we use $b = 0.75$ and $k_1 = 1.2$ which were reported as optimal for other IR collections [106].

6.5 Factoid QA Results

Table 6.3 summarizes the results of more than 700 experiments conducted over several feature combinations and LTR methods. Table 6.5 shows the p-values of statistical significance testing for differences between LTR models based on single feature sets.

Combination	**MART**	**Ranknet**	**RankBoost**	**LambdaMart**
POS	0.6587	0.5862	0.6730	0.6213
POS+D_TFIDF_1	0.6917 ↑ 5%	0.5953	0.6746	0.6200
POS+D_TFIDF_1+SOCIAL	0.7514 ↑ 14%	0.5931	0.6719	0.6361
POS+D_TFIDF_1+SOCIAL+WEMB_Q	0.7682 ↑ 17%	0.5946	0.6719	0.6464
POS+D_TFIDF_1+SOCIAL+WEMB_Q+D_TFIDF_3	0.7745 ↑ 18%	0.5904	0.6732	0.6204
POS+D_TFIDF_1+SOCIAL+WEMB_Q+D_TFIDF_3+ ... REPW	0.7788 ↑ 18%	0.5895	0.6733	0.6415
POS+D_TFIDF_1+SOCIAL+WEMB_Q+D_TFIDF_3+ ... REPW+TIME	**0.7795** ↑ 18%	0.5867	0.6755	0.6420

Table 6.3: Factoid task, best combinations of features sets, based on MRR@10, and their percent of improvement over the best single feature set (POS).

Single feature analysis. These results show that features obtained from the text of the messages (POS, WEMB_THR, CONTENT) yield good results compared to, for instance, relying solely on social signals such as replies, likes or retweets (SOCIAL). The single most predictive feature set for ranking answers to factoid questions is *Part-of-Speech* (POS), which significantly outperforms all the other features as shown in Table 6.4.

		MART		**Ranknet**		**RankBoost**		**LambdaMart**	
	Feature set	μ	σ	μ	σ	μ	σ	μ	σ
1.	POS ($^{2-12}$)	**0.6587**	0.0250	0.5862	0.0266	**0.6730**	0.0377	0.6213	0.0617
2.	WEMB_THR ($^{3-12}$)	**0.6202**	0.0296	0.5489	0.0315	0.6013	0.0264	0.5618	0.0388
3.	CONTENT ($^{4-12}$)	0.5763	0.0284	0.5694	0.0320	0.5543	0.0230	0.5900	0.0330
4.	D_TFIDF_1 ($^{9-12}$)	0.5282	0.0286	0.5299	0.0349	0.5143	0.0311	0.4966	0.0407
5.	SOCIAL ($^{8-12}$)	0.5280	0.0284	0.5490	0.0265	0.4766	0.0296	0.5311	0.0424
6.	D_WEMB (^{9,11,12})	0.5131	0.0303	0.5278	0.0313	0.5155	0.0337	0.5105	0.0341
7.	D_TFIDF_3 (^{9,11,12})	0.5123	0.0331	0.4057	0.0262	0.4716	0.0353	0.4075	0.0280
8.	D_TFIDF_2 (^{9,11,12})	0.5083	0.0303	0.4457	0.0277	0.4870	0.0315	0.4338	0.0254
9.	USERS	0.4857	0.0223	0.5344	0.0296	0.4883	0.0278	0.5376	0.0503
10.	WEMB_Q	0.4942	0.0258	0.4942	0.0258	0.4942	0.0258	0.4942	0.0258
11.	TIME	0.4815	0.0428	0.5150	0.0303	0.4942	0.0258	0.5560	0.0315
12.	REPW	0.4810	0.0326	0.3651	0.0347	0.4929	0.0328	0.4051	0.0677

Significant differences based on MART pairwise t-tests, $\alpha = .95$, Bonferroni correction.

Table 6.4: Factoid task MRR@10 results, mean (μ) and S.D. (σ). POS($^{2-12}$) means that POS is significantly better than feature sets 2 (WEMB_THR) to 12 (REPW).

Feature combination. Table 6.3 shows the results of several experiments combining feature sets in the LTR framework. The table shows the percent of improvement over the best

	D_TFIDF_1	D_TFIDF_3	POS	REPWORDS	SOCIAL	TIME
D_TFIDF_3 (0.5123)	0.10					
POS (0.6587)	0.00	0.00				
REPWORDS (0.4810)	0.00	0.00	0.00			
SOCIAL (0.5280)	1.00	0.02	0.00	0.00		
TIME (0.4815)	0.00	0.02	0.00	1.00	0.00	
WEMB_Q (0.4942)	0.00	0.07	0.00	0.61	0.00	0.63

Significant with $p < .05$

Table 6.5: P-values of multiple t-tests on MRR@10 over single isolated features, with Bonferroni correction. Numbers in parentheses show, for context, the mean MRR@10 of the feature using MART LTR model.

performing feature set POS (MRR@10 = 0.6587), and we show that a combination with content (D_TFIDF_1, WEMB_Q, D_TFIDF_3, REPW), social and time feature sets can increase the performance up to 18.3% (MRR@10 = 0.7795), showing that these features provide different types of signals for the ranking task.

Feature contribution. As mentioned, we use an additional metric to evaluate the contribution of our set of features, called EFE. Figure 6.6 shows the contribution of each feature. It means social features are very helpful to determine the relevance and might be less correlated than other pairs of features. In contrast, the contribution of the time feature is the lowest. However, we cannot eliminate features because each of them contributes something that others have not yet incorporated.

Set of feature contribution	EFE
SOCIAL	**8.28%**
D_TFIDF_1	7.38%
POS	3.45%
WEMB_Q	2.18%
D_TFIDF_3	1.13%
REPW	0.58%
TIME	0.47%

Table 6.6: Contribution of each set of features regarding the best combination. We use the metric EFE proposed by [94].

About the LTR Methods. Considering both evaluations –on each feature set and over combinations– the best method is MART, especially in the feature set combination results of Table 6.3. Although LambdaMart is usually presented as the state of the art, there is also recent evidence on non-factoid QA showing MART as the top performing algorithm [116], in line with our results. Notably, all the methods show a strongly correlated behavior in terms of feature set ranking. Moreover, these methods presented the best MRR@10 results using the POS feature (Table 6.4). In contrast, the REPW presented poor results (with the exception of RankBoost). This consistent behavior underpins our conclusions in terms of the importance of POS for this task.

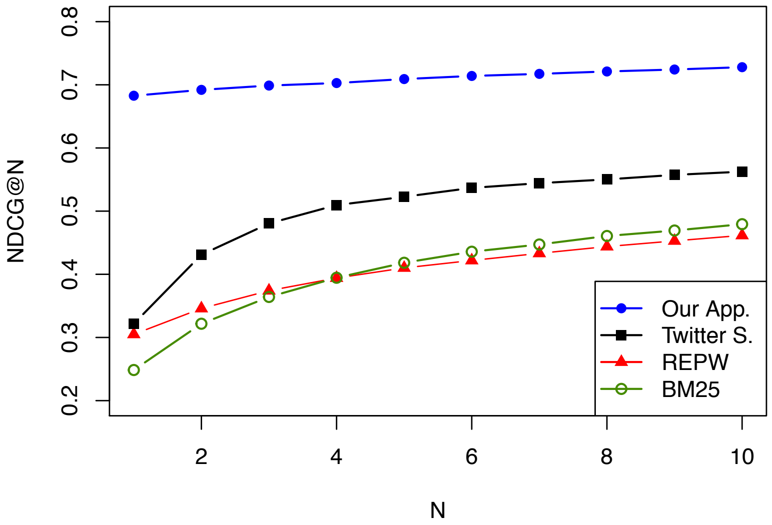


Figure 6.1: Ranking quality comparison between the baselines and the best feature combination using MART algorithm.

Baselines results. Table 6.7 shows the MRR@10 of our best method and the baselines. Our best combined LTR model beats all baselines, improving the factoid ranking results by 74.77% in terms of MRR@10 and by 29.4% on nDCG@10 over Twitter Search.

Method	**MRR@10**	**nDCG@10**
BM25	0.3852	0.4793
Twitter Search	0.4460	0.5625
REPW	0.4810	0.4616
Best comb.	**0.7795**	**0.7279**

Table 6.7: Results of our best combination vs baselines.

In addition, Figure 6.1 shows the ranking quality of baselines and our method from 1 up to 10 positions. Our method performs well from the beginning. The other methods started with a poor performance and then improved bottom ranking positions. We observe, that Twitter Search starts with poor performance, but then (at 2nd ranking position) increases quality and performs better than the other baselines. It means that the ranking method of Twitter Search performs well despite our not knowing precisely how it works. However, our method performs better than all other instances from 1 to 10.

About random collection. Given that one of our major concern was to avoid the overfitting problem, we applied two methods for evaluation; cross validation and our 30 randomized collections. Table 6.8 shows the MRR@10 results using both methods. As we see, results shows low difference between them. It means that we could use any of them.

	MRR@10 (MART)			
	CV=3	CV=5	CV=7	30 rand col.
POS	0.6410	0.6580	0.6388	0.6587
POS+D_TFIDF_1	0.6871	0.6912	0.6868	0.6917
POS+D_TFIDF_1+SOCIAL	0.7481	0.7495	0.7466	0.7514
POS+D_TFIDF_1+SOCIAL+WEMB_Q	0.7501	0.7597	0.7674	0.7682
POS+D_TFIDF_1+SOCIAL+WEMB_Q+D_TFIDF_3	0.7592	0.7680	0.7598	0.7745
POS+D_TFIDF_1+SOCIAL+WEMB_Q+D_TFIDF_3+REPW	0.7612	0.7624	0.7543	0.7788
POS+D_TFIDF_1+SOCIAL+WEMB_Q+D_TFIDF_3+REPW+TIME	0.7682	0.7692	0.7633	0.7795

Table 6.8: MRR@10 of our best combinations (Table 6.3) using cross validation with 3,5 and 7 folds, and the best MRR@10 using our method of 30 random collections. In all instances, the differences are lower.

6.6 Factoid QA Characteristics

We found out special characteristics of questions and threads in our factoid QA dataset; described as follows:

Average Arrival Time of Replies. We found that the average time between replies was low in long threads. In particular, we found 1,525 threads with more than 5 replies. In 85.9% of these threads, the average arrival time for replies was 2 hours. This means that the replies arrive quickly in long threads (long discussions). Figure 6.2 shows this behavior.

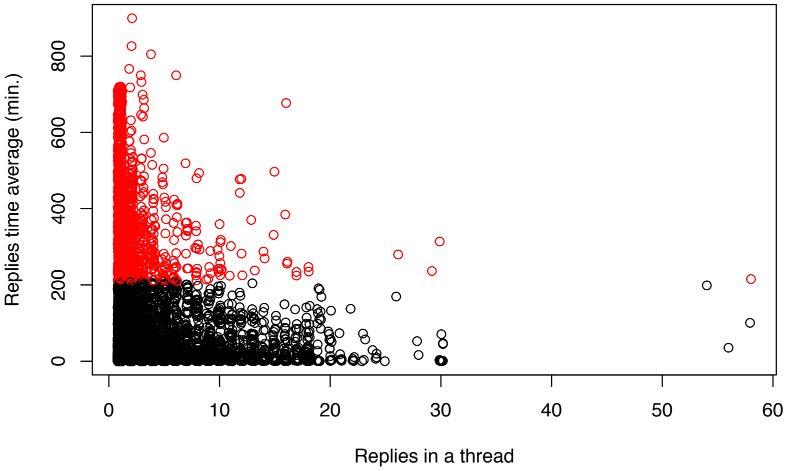


Figure 6.2: The time average between replies of long threads (≥ 5 replies) tends to arrive in a short time frame.

In a future work, we could identify these kinds of questions that generate long discussions and estimate the amount of replies.

Query Formulations (QF). We study the relation between the QF and the relevance of threads retrieved. Table 6.9 shows the amount of relevant/ non-relevant threads retrieved for

Query Formulations	Relevant	non-relevant
QF_1	0	6
QF_2	2	15
QF_3	8,421	20,392
QF_4	379	697
QF_1-QF_2	1	10
QF_2-QF_4	8	26
QF_3-QF_4	397	1,805
QF_1-QF_2-QF_4	25	84
QF_2-QF_3-QF_4	84	239
QF_1-QF_2-QF_3-QF_4	89	1,193
	9,406	24,467

Table 6.9: Amount of relevant and non-relevant threads retrieved per QF. For example, using QF_2, we retrieved 2 relevant and 15 non-relevant threads; for QF_3-QF_4 (meaning that a thread was retrieved using these two QF), we retrieve 397 relevant and 1,805 non-relevant threads. NOTE: We omit QF's that we do not retrieved any tweet/thread (for example, QF_1-QF_3).

each QF. In this inspection, QF_3 and QF_4 retrieve more threads. The combination of QFs, for example QF_1-QF_2, means that a thread was retrieved using these two QF. In these cases, we consider just one instance. We expected that these QF are most useful because they eliminate noisy words such as stopwords and non-alphanumeric characters. In particular, QF_3 retrieved more similar threads related to questions (because it includes 6W1H) and, QF_4 retrieved more general threads (without 6W1H). In fact, we could use just these two for the retrieving phase.

The distance feature. The best combination (Table 6.3) indicates that computing distance with 1, 2 and 3-grams performs well using TFIDF vectors. In contrast, word embeddings have a poor performance in the distance feature. Likely, word embeddings do not seem appropriate for these distance metrics.

Likewise, word embeddings perform well when are they used as an explicit vector of questions or threads. This is an important issue because there are studies, such as Molino et al. [78], where they use embeddings to compute distance metrics (similar to our work), and they did not perform well. They likely could have had better results using the explicit vector representation of the questions and answer feature embeddings.

POS with {2-3}-grams. We addressed additional experiments in order to analyze whether another configuration of POS could improve the ranking process. We study the contribution of POS considering also 2 and 3 grams (bigrams and trigrams, respectively). We determine the top-30 bigrams and trigrams of relevant threads. We then, run experiments using MART with our POS feature, POS_BI (bigrams) and POS_TRI (trigrams) and the combination between them. Table 6.10 shows the results of these experiments. The contribution using those three features together is not significant compared with the use of single POS (it increases by only 0.91%). For that reason, we consider that computationally, it is not convenient to use bigger n-grams of POS as a new feature.

	Features	MRR@10 (MART)
(1)	POS (from Table 6.3)	0.6587
(2)	POS_BI	0.6170
(3)	POS_TRI	0.5806
(4)	POS+POS_BI	**0.6635**
(5)	POS+POS_TRI	0.6630
(6)	POS+POS_BI+POS_TRI	**0.6647**
(7)	POS_BI+POS_TRI	0.6242

Table 6.10: If we fix POS, the contribution of POS_BI -bigram- is slightly higher than POS_TRI (trigram); (4) and (5), respectively. However, despite that these three features combined present better performance (6), the contribution is not significant (0.91%).

Participants and time to get replies. Table 6.11 shows an evaluation of the characteristics inside the threads. We evaluate three aspects in the threads; *(1) the user participation*: the number of different users that participate in the thread, *(2) the time needed to receive the first reply*: time difference between the initial tweet and the first reply and, *(3) the duration of threads*: time difference between the initial tweet and the last reply.

	Range	Median
(1) Number of Participants	1-21	2
(2) Time of the first reply	1 sec - 24 hours	12.8 min
(3) Duration of threads	1 sec - 24 hours	42.1 min

Table 6.11: Characteristics of conversation threads regarding user participation and time.

Honey et al. [48] report a similar number of participants (1) in their study of @mentions in Twitter. In particular, they report a range of $2-10$ users per thread and we report $1-21$.

Regarding (2), we found that in 62% of cases, the first reply of threads takes less than 30 min and 93% take less than 10 hours. Sharoda et al. [98] reports similar values in their study; 67% of questions receive the first reply within 30 min and 95% within 10 hours. Moreover, they report a median of 10.3 min to receive the first reply and we report 12.8 min. In short, observing the median of the time to receive the first reply of any thread is around 13 min.

Finally, we report the duration of threads (3). Honey et al. [48] address a similar study with different results. They report a median of 26.33 min in comparison to our 42.1 min. The difference may be due to them considering @mentions as conversations and it does not always represent a conversation thread. In our study, we inspect real conversations as threads taken directly from Twitter. They likely retrieved fewer data than us.

In short, taking the median, conversation threads in Twitter are completed (or satisfied) in around 42 min. After that time, threads are more likely to stop receiving replies. However, the range is wide and could takes up to 24 hours.

Time for the first reply. We repeat the previous analysis but now dividing relevant and non-relevant threads. We focus on the first reply of a thread in order to understand whether relevant threads are more likely to receive sooner replies or not. We observed that the first reply of relevant threads takes longer to arrive compared to non-relevant ones. Hence, time seems to be a determining factor of relevance. In fact, if we observe the previous analysis of the arrival time of the first reply of any thread (Table 6.11), the result is 12.8 mins and it is very similar to the non-relevant case. This means that relevant threads take longer (about 4 min of difference) to receive the first reply.

6W1H's Contribution. We study the contribution of 6W1H in thread relevance. Table 6.12 shows the number of questions that were answered by using our 6W1H definitions. The majority of relevant threads were asked using "who", "where", "what" y "when". Questions with "how" and "which" do not generate relevant threads, because they are not precise and trigger different kinds of replies which require a semantical interpretation (despite that they are factoid). For instance, "**How** did James Dean die?", the correct answer is "in a car crash", but other valid answers could have been: "in a car accident" or "driving". In contrast, question as "**Who** invented the paper clip?" has a concrete answer: "Johan Varler".

6W1H	**# Quest.**	**# Threads**	**# Rel**	**# Non-Rel**	**% Rel**
Who	95	7,306	2,465	4,841	33.7%
Where	61	4,410	1,260	3,150	28.6%
What	217	12,265	3,403	8,862	27.7%
When	72	6,742	1,856	4,886	27.5%
How	37	2,487	283	2,204	11.4%
Which	9	619	137	482	22.1%
Why	0	0	0	0	0%

Table 6.12: Amount of threads retrieved based on 6W1H. Note: We omit 44 threads whose questions did not contain 6W1H.

In short, our method (factoid QA), retrieves more relevant threads when questions are about "what", "who", "where" and "when".

Final Comments. This analysis demonstrates how conversation threads are posed in Microblogs, in particular, Twitter. One aspect is about the importance of the type of questions in order to receive relevant answers. Questions that ask about people (**Who**) or places (**Where**), are more likely to receive relevant answers (likely because they are easy to answer them).

Also, we proposed four query formulations (QF) which are the key to retrieving answer candidates. Two of them (QF_3 and QF_4) are most useful for the retrieval phase, because they retrieve more similar threads related to questions and hence are more likely to find correct answers.

With respect to time, threads with a lot of replies are more likely to get answered sooner. However, we expected to receive lower participation because they are factoid questions.

Finally, we compare our results with other similar approaches. The main differences is that we inspected conversation threads in Twitter and not just tweets as mentions (@mentions). This is a very important difference because our results focus on real conversations carried out in Microblogs.

In the next section, we use our best LTR model for non-factoid questions.

Chapter 7

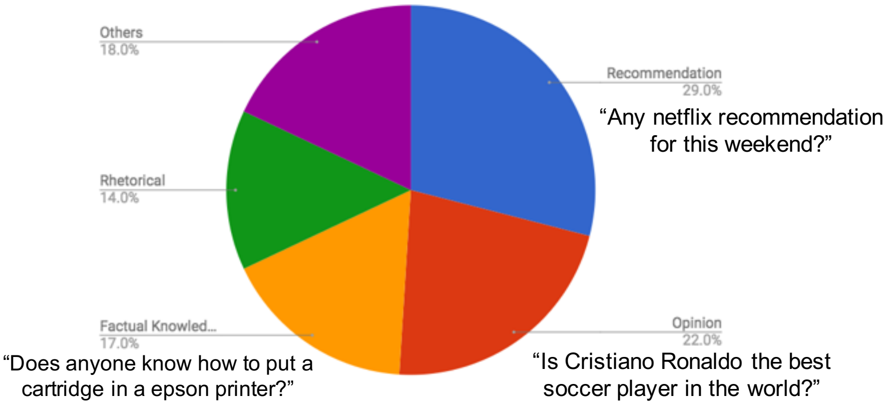
Non-factoid QA Task

In previous chapters, we demonstrate that historical microblog data can be used to answer factoid questions. However, there are other types of questions: non-factoid QA. These questions require more than one answer such as *"can anyone recommend a good restaurant in NY?"* unlike factoid QA where there is one answer (e.g., *What year dead Pablo Neruda?*). People ask these kinds of questions in microblogs because their answers are related to giving recommendations, opinions, suggestions, among others [80]. Our intention is to use the factoid task as a proxy to our goal of answering more complex questions (non-factoid) by employing transfer learning, i.e., learning a model for one task (factoid QA) and transferring the model for a different task (non-factoid QA.)

In this Chapter, we evaluate non-factoid questions using the best trained model of Chapter 6 In particular, we focus on non-factoid questions related to recommendations taken directly from the microblog Twitter. Furthermore, we study in further depth recommendation tweets posted on Microblog Twitter which are not necessarily paraphrased as questions. Our results show that we are able to answer these more complex questions using transfer learning (**RQ3**). In fact, we can improve the ranking by removing tweets with low content. The results from this chapter were published in the conference ECIR 2018 [45].

7.1 Non-factoid QA

Factoid QA is only one type of task, and recently it has been successfully tacked using deep learning models [4, 5, 49, 96, 110, 119]. Since our final goal is leveraging all the contextual variables available in microblogging for QA –recency, questions with spatio-temporal context, etc.– beyond factoid QA, we also explore the generalization of the methods and features already analyzed towards non-factoid QA. Unlike factoid QA, non-factoid QA questions have more than one answer and are usually associated with questions that require opinions, recommendations, experiences, among similar others. Two examples of these kinds of questions are: "Anyone have any GIF maker software or sites they can recommend?" and "Anyone have a remedy for a headache?". In particular, we focused on questions related to recommendations which are other types of questions that users request [56, 80, 98]. Morris et al. [80]



“Any netflix recommendation for this weekend?”

“Does anyone know how to put a cartridge in a epson printer?”

“Is Cristiano Ronaldo the best soccer player in the world?”

Figure 7.1: Question types that people ask in social networks (by [80]).

report that these questions account for a large proportion of questions in Twitter (around 30%). Figure 7.1 shows the mostly asked question types in social networks. We made a more in depth analysis of these types of questions in the Appendix A1.

One important issue with non-factoid QA is the lack of datasets to build a ground truth for training and testing in the way we addressed the factoid QA task. We then explore this type of QA through transfer learning, which is a strategy in machine learning “*motivated by the fact that people can apply knowledge learned previously to solve new problems*” [84]. In our case, we collect a small dataset of non-factoid questions and answers and test our existing QA factoid model towards this new task. Our results show that we are able to answer more complex questions using transfer learning, and we provide all the details in this section.

7.2 Data Collection.

We first collect data from Twitter and we search recommendation questions using the query “`recommend* ?`”. Figure 1.1 shows two non-factoid threads retrieved using this query.

For the preliminary evaluation of our approach, we sampled 40 diverse non-factoid questions taken from Twitter. Unlike the Factoid QA dataset of Chapter 6, we do not have the ground truth of correct answers. Note that the goal of this chapter is to evaluate whether our model performs well in other types of questions. In particular, our model should be able to predict the ranking of a set of candidate threads. In fact, LTR framework works in that way (Figure 2.1).

The difference in sizes of factoid and non-factoid datasets, shown in Table 7.1, justifies transferring our existing model, rather than learning a new one from non-factoid data. Unlike the factoid questions dataset, we do not have the ground truth of correct answers. In fact, the correct answer now could be more than one since we are inspecting non-factoid questions (replies average in factoid and non-factoid QA is 0.9 and 3.32, respectively). This is because users are encouraged to participate in giving their recommendations, advising or discussing experiences. In contrast, factoid answers are sooner in threads (answers are usually in a

	non-factoid	**Factoid**
Number of questions	40	491
Number of tweets	2,666	63,646
Number of threads	386	33,873
% tweets that are part of a thread	87.99%	46.70%
Avg. replies per thread	332	0.9

Table 7.1: Datasets description of non-factoid and factoid QA. Size differences justify the need for transfer learning.

simple tweet or in first positions of a thread). We therefore manually inspected and evaluated the top-10 answers ranked with our approach for each of the 40 questions, labeling them as relevant and non-relevant (following the same definition of relevant and non-relevant threads of factoid QA).

It was necessary to apply additional filtering in some questions. In the case of factoid question tasks, we use well-formulated questions from TREC and a curated dataset. But at this time, we retrieve questions directly from Twitter and these are usually mis-written or even present irrelevant information. Therefore, before computing the query formulations of the pipeline, in some questions we apply an additional filtering for removing messages which contain:

- Text as @mentions, #hastags or thanks.
- Irrelevant phrases as “please, can anyone...” or “anyone of my friends...”.
- Irrelevant adjectives such as “good”, “cheap”, “fabulous”, “near”. For example, “can anyone recommend me a **cheap** hotel near the star theatre in singapore?”, “What fun, **fabulous** restaurant do you recommend in Vegas for dinner, please?”, “can anyone recommend me a **cheap** hotel **near the star theatre** in singapore?”.

  We will use these adjectives to evaluate the strength of questions in a future project.

Therefore, we manually inspected and evaluated the top-15 answers ranked with our approach for each of the 40 questions, labeling them as relevant and non-relevant .

7.3 Experimental Setup

For each query, we apply the query formulations (QF) and build the threads using the Twitter API. We then extract just the set of features of the best combination (Table 6.3) which corresponds to POS, D_TFIDF_1, SOCIAL, WEMB_Q, D_TFIDF_3, REPW and TIME.

We generate a ranking of threads using our trained model of the best combination in Factoid QA Task. More specifically, we used transfer learning [84, 85, 107] (i.e., our best factoid QA LTR model) to rank answers (threads) for the non-factoid task.

Figure 7.2 shows the pipeline that we follow in order to test non-factoid questions in our

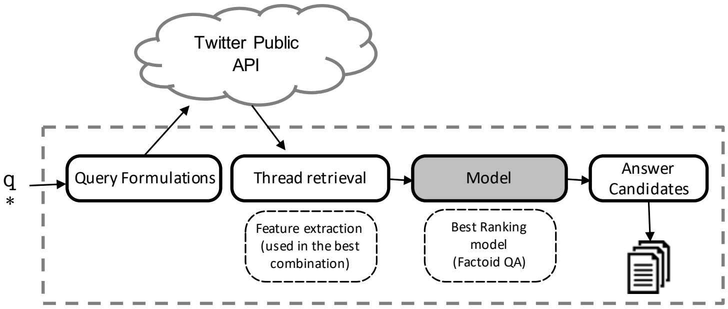

Twitter Public API
q*
Query Formulations
Thread retrieval
Model
Answer Candidates
Feature extraction (used in the best combination)
Best Ranking model (Factoid QA)

Figure 7.2: A pipeline used for non-factoid questions. Given a question q^*, we retrieve and build the threads using query formulations and Twitter API. We extract just the features of the best combination and evaluate using the trained ranking model. The output are the answer candidates (threads).

trained model. Essentially, we apply the same basic steps of the pipeline of Figure 5.1, but we adapt it for new questions. Therefore, the input are new questions ($q*$) and the output are the answers (threads).

7.3.1 Manual Evaluation

We annotate manually features of threads studied on Chapter 6. For each thread, we check whether there were answers inside threads or not. Since we are not experts in all topics (e.g. hotels or restaurants in unknown cities, anime movies, books recommendation, among others), we employed valid web sources to ensure answers are correct.

In addition, we annotate other characteristics to evaluate the quality of the results. In particular, we explore ideas from related studies about answer quality addressed in QA. For instance, Jeon et al. [52] studied non-textual features of answers in Naver Q&A (a Korean QA portal). They considered characteristics to predict the quality of answers, such as the answer length, user activity level and number of answers, among others. Shah et al. [97] addressed a similar study, but evaluating features of answers from Yahoo Answers. Through a human evaluation, they studied 13 different criteria to measure quality of answers. For example, if answers are informative (provide enough information), polite (offending degree), readable and other similar features of [52].

We take the informative metrics of the related studies mentioned above and adapt them to our problem. Although our approach is not directly related to the quality of questions or answers in QA with quality, we adapt the metrics to the context of our study. We define two specific heuristics to evaluate the answers[1].

(1) If some tweet in a thread is a **direct answer** to the initial question, it means that the answer is written explicitly inside the thread.

(2) If some tweet in a thread contains an **indirect answer**, i.e., the correct answer is not written explicitly but there is a reference to other sources (such as a URL or a mention

[1] In this analysis, a **reply** is a tweet which is part of a thread (without counting the initial tweet). Figure 1.1 shows the parts of a thread.

Initial Question: *Can Anyone Recommend a great hotel in Barcelona? 4 daybreak. Thanks.*			
THR 1	**Init.tweet:**	-	Hey guys thinking to visit Barcelona. Anyone can recommend great hotel in the centre and places to visit? Planning to visit with my mom.
	R1:	(DA)	Novotel Barcelona City is in the heart of the city on Avenida Diagonal, one of the city's main streets.
THR 2	**Init.tweet:**	(DA)	Btw, if you're heading to Barcelona I definitely recommend the Ohla hotel. Great location, staff, rooms, food `http://t.co/41rB2EKFRE`.
THR 3	**Init.tweet:**	(DA)	Great vacation in Spain. Highly recommend El Palace Hotel in Barcelona (you have to tell cabbies it used to be The Ritz).
	R1:	-	Looking into city break ideas. So far on the list I have Copenhagen, Barcelona and Lisbon. Anywhere else I should be looking?
	R2:	-	Barcelona all the way, I even have a great hotel recommend actually I'll Facebook you...
	R3:	-	Ahhh amazing! Thank you lovely.
THR 4	**Init.tweet:**	(IA)	We totally recommend @axelfriendly hotel in Barcelona and we had a great visit by Maxi `http://t.co/q9ZfP9C7j4`.
THR 5	**Init.tweet:**	-	Still looking for a hotel if anyone has any recommendations?

Table 7.2: An example of threads retrieved by our method which could answer the initial question. Each tweet of threads can be a direct answer (DA), an indirect answer (IA) or a tweet which does not answer the initial question (-).

to another user with "user").

For instance, Table 7.2 shows five thread candidates retrieved by our method and which can be answer the initial question: *Can Anyone Recommend a great hotel in Barcelona? 4 daybreak. Thanks.*. We describe each thread as follows:

- Thread 1 has an initial tweet and one reply (R1), and the latter is a **direct answer** (underlined), because it is correct (literal) and valid.
- Thread 2 does not have any replies (we denoted as **thread without replies**), but the initial tweet of the thread contains a **direct answer**.
- Thread 3 have three replies (we denoted as a **thread with replies**), but present one **direct answer** (DA) in the initial tweet.
- Thread 4 is a **thread without replies** and it presents an **indirect answer** (IA), because the answer does not appear explicitly in the tweet (likely inside the URL).
- Thread 5 is a **thread without replies** and also it does not have direct or indirect answer.

We carried out a manual evaluation to validate the relevance of the answers, either direct or indirect.

7.3.2 Results

Table 7.3 shows the results of the evaluation. We annotate and evaluate the ranking position of direct and indirect answers. Items 1 to 3 show that there are few threads without replies that incorporate answers. In addition, observing the average replies per threads (Item 7) we notice that it is larger than for factoid evaluation; 0.9 for our factoid dataset and 5 for

			Average
Threads per question without replies:			
1)	...	and without direct answers.	4.98
2)	...	with direct answers (or single tweet).	1.15
3)	...	with indirect answers.	0.41
Ranking position of:			
4)	...	the thread with the first direct answer.	2.69
5)	...	replies (inside threads) with the first direct answer.	1.33
6)	...	replies (inside threads) with first indirect answer.	0.87
Replies of a thread:			
7)	...	per thread.	5.00
8)	...	with direct answers.	1.72
9)	...	with indirect answers.	0.56
10)	...	with direct answers and review.	1.10
11)	...	with direct answers, review and URL.	0.11
12)	...	with direct answers with a helpful URL.	0.23
Threads			
13)	...	without replies, without direct answers and before the first thread with direct answer.	1.50

Table 7.3: Analysis of top-15 threads ranked by our model of non-factoid questions.

non-factoid. We can explain this difference considering that non-factoid are more likely to generate more replies (especially in this case of recommendation questions). The position of the first direct answer (Item 4) comes up in the 2nd or 3rd thread, on average.

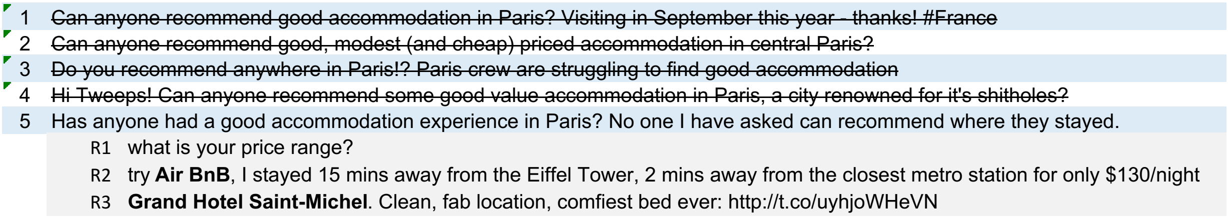
Regarding the thread replies, the position of the first direct answer occurs generally in the first positions (Items 5 and 6). We also found that around two replies per thread (on average) contain direct answers (Item 8) and around one indirect answer (Item 9). In fact, we found threads without replies (just tweets) but with indirect answers. Moreover, around one reply of a thread (on average) contains a review, experience or additional information related to the direct answer (Item 10).

Surprisingly, we found that a low amount of replies contain just URLs for additional or helpful information (Item 12). In fact, few replies contained these three elements; direct answer, a review and URL (Item 11).

MRR. We obtained a $MRR@10 = 0.5802$, which is good compared to results reported recently (MRR=[0.4-0.45] in [116]), but lower what we obtained in the factoid QA task.

Enhancing MRR. By further analyzing the data we found that, on average, for every question we retrieved, 1.5 threads without any reply were also non-relevant to the question made.

We propose a strategy to improve the MRR. Observing the value of Item 13, there are 1.50 threads without replies on average before the first thread with a direct answer. Based on this, we discard those threads from potential answers without replies (or just single tweets). Figure 7.3 features four of five threads that are removed because they do not have replies. Notice how thread number 5 is moved up to the first position, improving the MRR.



1 ~~Can anyone recommend good accommodation in Paris? Visiting in September this year - thanks! #France~~
2 ~~Can anyone recommend good, modest (and cheap) priced accommodation in central Paris?~~
3 ~~Do you recommend anywhere in Paris!? Paris crew are struggling to find good accommodation~~
4 ~~Hi Tweeps! Can anyone recommend some good value accommodation in Paris, a city renowned for it's shitholes?~~
5 Has anyone had a good accommodation experience in Paris? No one I have asked can recommend where they stayed.
R1 what is your price range?
R2 try **Air BnB**, I stayed 15 mins away from the Eiffel Tower, 2 mins away from the closest metro station for only $130/night
R3 **Grand Hotel Saint-Michel**. Clean, fab location, comfiest bed ever: http://t.co/uyhjoWHeVN

Figure 7.3: An example of five candidate threads. Thread number 5 has three replies with one indirect answer (R1) and one direct answer (R3). If we consider all threads (without any thread removed), the MRR is 0.2. Removing the first four threads without replies (crossed out text) the MMR is 1.

We are aware that the proposed solution implies the lost of potentially relevant information but this strategy improved the results to $MRR@10 = 0.6675$, with the small trade-off of one question out of 40 for which we could not find answers. This happens because there are threads without replies but with direct answers. Likewise, the MRR improved by 15% applying the removal process (Table 7.4). Therefore, this method gives importance to threads that have replies and it seems to be reasonable for recommendation-based non-factoid QA.

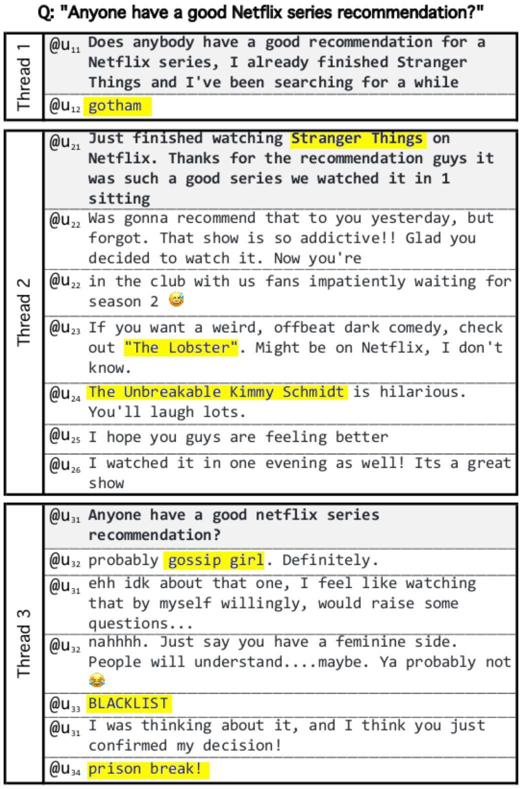
Case	MRR	nDCG
Our method using non-factoid dataset	0.5802	0.6784
... removing threads without replies	**0.6675**	0.7566

Table 7.4: The MRR@10 and nDCG@10 of our approach and removing threads without replies. For the latter we have to eliminate 3 questions where all the threads were removed.

Unlike our factoid QA dataset, non-factoid QA tends to have more replies inside threads (Table 7.1). We can explain this because users are encouraged to participate giving their recommendations, advising or telling experiences. In contrast, factoid answers are concrete and show up earlier in threads (answers are usually in a simple tweet or in top ranked positions of a thread). In fact, the average number of replies for factoid and non-factoid tasks is 0.9 and 3.32, respectively.

7.4 Non-factoid examples using our approach

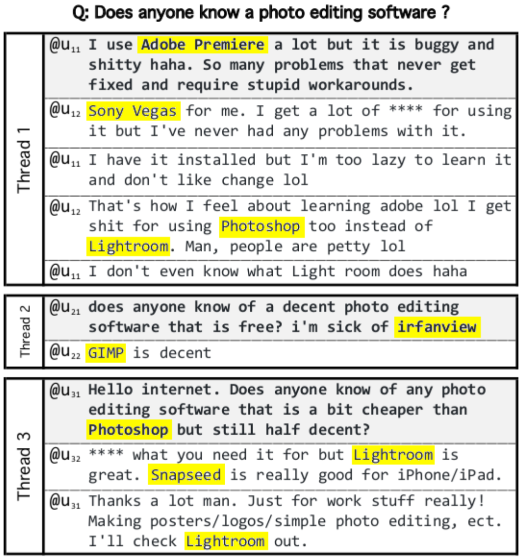
Figures 7.4, 7.5 and 7.6 show examples where we use our approach to retrieve answers (threads/tweets). In each case, given an initial question (query), we show the top threads that our approach left at the top of the list.



Q: "Anyone have a good Netflix series recommendation?"

Thread 1	@u$_{11}$	Does anybody have a good recommendation for a Netflix series, I already finished Stranger Things and I've been searching for a while
	@u$_{12}$	gotham
Thread 2	@u$_{21}$	Just finished watching Stranger Things on Netflix. Thanks for the recommendation guys it was such a good series we watched it in 1 sitting
	@u$_{22}$	Was gonna recommend that to you yesterday, but forgot. That show is so addictive!! Glad you decided to watch it. Now you're
	@u$_{22}$	in the club with us fans impatiently waiting for season 2 😅
	@u$_{23}$	If you want a weird, offbeat dark comedy, check out "The Lobster". Might be on Netflix, I don't know.
	@u$_{24}$	The Unbreakable Kimmy Schmidt is hilarious. You'll laugh lots.
	@u$_{25}$	I hope you guys are feeling better
	@u$_{26}$	I watched it in one evening as well! Its a great show
Thread 3	@u$_{31}$	Anyone have a good netflix series recommendation?
	@u$_{32}$	probably gossip girl. Definitely.
	@u$_{31}$	ehh idk about that one, I feel like watching that by myself willingly, would raise some questions...
	@u$_{32}$	nahhhh. Just say you have a feminine side. People will understand....maybe. Ya probably not 😂
	@u$_{33}$	BLACKLIST
	@u$_{31}$	I was thinking about it, and I think you just confirmed my decision!
	@u$_{34}$	prison break!

Figure 7.4: Query: Anyone have a good Netflix series recommendation? Possible answers are highlighted as yellow.

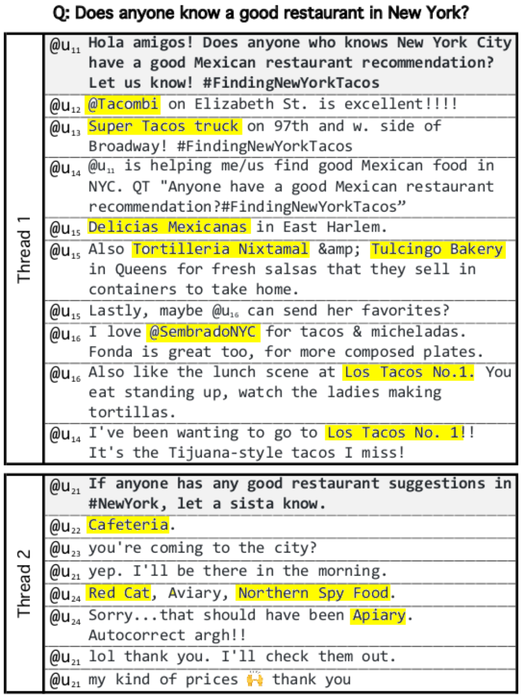


Q: Does anyone know a photo editing software ?

Thread 1	@u$_{11}$	I use Adobe Premiere a lot but it is buggy and shitty haha. So many problems that never get fixed and require stupid workarounds.
	@u$_{12}$	Sony Vegas for me. I get a lot of **** for using it but I've never had any problems with it.
	@u$_{11}$	I have it installed but I'm too lazy to learn it and don't like change lol
	@u$_{12}$	That's how I feel about learning adobe lol I get shit for using Photoshop too instead of Lightroom. Man, people are petty lol
	@u$_{11}$	I don't even know what Light room does haha
Thread 2	@u$_{21}$	does anyone know of a decent photo editing software that is free? i'm sick of irfanview
	@u$_{22}$	GIMP is decent
Thread 3	@u$_{31}$	Hello internet. Does anyone know of any photo editing software that is a bit cheaper than Photoshop but still half decent?
	@u$_{32}$	**** what you need it for but Lightroom is great. Snapseed is really good for iPhone/iPad.
	@u$_{31}$	Thanks a lot man. Just for work stuff really! Making posters/logos/simple photo editing, ect. I'll check Lightroom out.

Figure 7.5: Query: Does anyone know a photo editing software? Possible answers are highlighted as yellow.

However, there are cases where the approach fails. These cases are mainly due to misspelled texts (in questions and/or answers), poor keywords for the searching process, incomplete/missing information or the information is based on some context (for example, localization). Recall that our QF (query formulation) is based on filter keywords. Figure 7.7 shows an example where the question has irrelevant keywords such as *family*, *thinking*, *going*, *out*.

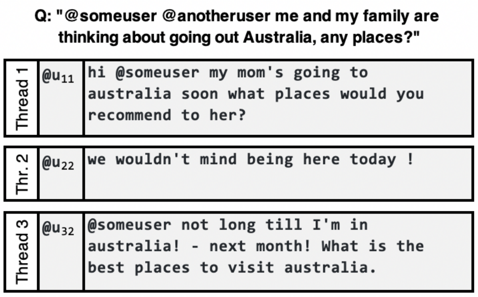


Q: Does anyone know a good restaurant in New York?

Thread 1	@u_{11}	Hola amigos! Does anyone who knows New York City have a good Mexican restaurant recommendation? Let us know! #FindingNewYorkTacos
	@u_{12}	@Tacombi on Elizabeth St. is excellent!!!!
	@u_{13}	Super Tacos truck on 97th and w. side of Broadway! #FindingNewYorkTacos
	@u_{14}	@u_{11} is helping me/us find good Mexican food in NYC. QT "Anyone have a good Mexican restaurant recommendation?#FindingNewYorkTacos"
	@u_{15}	Delicias Mexicanas in East Harlem.
	@u_{15}	Also Tortilleria Nixtamal &amp; Tulcingo Bakery in Queens for fresh salsas that they sell in containers to take home.
	@u_{15}	Lastly, maybe @u_{16} can send her favorites?
	@u_{16}	I love @SembradoNYC for tacos & micheladas. Fonda is great too, for more composed plates.
	@u_{16}	Also like the lunch scene at Los Tacos No.1. You eat standing up, watch the ladies making tortillas.
	@u_{14}	I've been wanting to go to Los Tacos No. 1!! It's the Tijuana-style tacos I miss!
Thread 2	@u_{21}	If anyone has any good restaurant suggestions in #NewYork, let a sista know.
	@u_{22}	Cafeteria.
	@u_{23}	you're coming to the city?
	@u_{21}	yep. I'll be there in the morning.
	@u_{24}	Red Cat, Aviary, Northern Spy Food.
	@u_{24}	Sorry...that should have been Apiary. Autocorrect argh!!
	@u_{21}	lol thank you. I'll check them out.
	@u_{21}	my kind of prices 🙌 thank you

Figure 7.6: Query: Does anyone know a good restaurant in New York? Possible answers are highlighted as yellow.

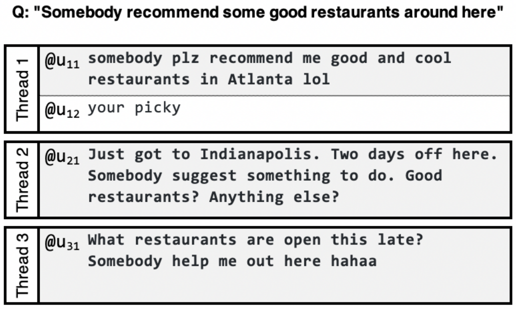
Figure 7.8 shows another case in which our approach does not provide accurate results. In this case the question is dependent on the location of the user.



Q: "@someuser @anotheruser me and my family are thinking about going out Australia, any places?"

Thread 1	@u_{11}	hi @someuser my mom's going to australia soon what places would you recommend to her?
Thr. 2	@u_{22}	we wouldn't mind being here today !
Thread 3	@u_{32}	@someuser not long till I'm in australia! - next month! What is the best places to visit australia.

Figure 7.7: A case where the proposed approach fails to provide good results, this is due to noisy keywords which select irrelevant information for the question, such as *thinking* and *family*.



Q: "Somebody recommend some good restaurants around here"

Thread 1	@u_{11}	somebody plz recommend me good and cool restaurants in Atlanta lol
	@u_{12}	your picky
Thread 2	@u_{21}	Just got to Indianapolis. Two days off here. Somebody suggest something to do. Good restaurants? Anything else?
Thread 3	@u_{31}	What restaurants are open this late? Somebody help me out here hahaa

Figure 7.8: A case where the approach fails due the localization context dependency of the question.

Chapter 8

Discussion

We have studied and demonstrated that microblogs contain valuable information that can be leveraged to obtain answers for information needs paraphrased as questions. Our experimental results validate the potential for using microblog data for factoid and non-factoid QA, identifying the most informative features as well as the best LTR model.

In this section, we discuss the results of our research. We focus on the most important aspects of our findings such as the best features, transfer learning and LTR methods.

Feature importance in microblogs QA. One of the most interesting findings of our evaluation is the high predictive power of POS features. Previous work on community QA conducted on a large dataset of Yahoo! Answers [78] found similar results where proper nouns (such as names of places, people, etc.) and prepositions (such as at, before, on, etc.) are good predictors for relevant answers. It makes sense, especially for factoid QA.

In addition, we found a good discriminative effect of coordinating conjunctions (and, but, or, so), and we recommend not removing these features as "stop words" before conducting POS tagging in factoid QA using microblog data. Figure 8.1 summarizes and present the most relevant tags of POS using our factoid QA dataset.

However, we discovered some important differences with the results in Molino et al. [78]. They found punctuation as a discriminative feature between relevant and non-relevant answers, whereas it was not helpful in our case. Most likely, this result is explained by users of traditional QA platforms, who typically tend to write longer answers compared to microblogs. This phenomena might occur due to space constraints of microblog platforms.

Regarding the contribution of each single feature, Figures 8.2 and 8.1 show the most important features for discerning thread relevance. Surprisingly, social features of Twitter are not good predictors of relevance, with the exception of hashtags. In an inspection of our dataset, we found that threads with hashtags (e.g. #superbowl, #NYC, etc.) are, in general, questions that users want to spread beyond his/her network, hence they could tend to receive correct answers. For example, given the question "How many floors are in the Empire State

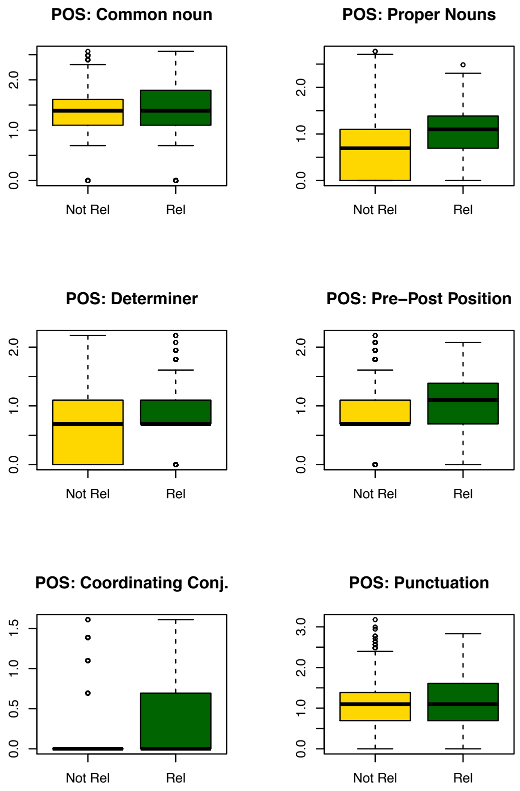


Figure 8.1: Part-of-speech (POS) features that discern among relevant and non-relevant threads. Examples: Common noun: {moon, tree}, Proper noun: {California, Google}, Determiner: {the, a}, Pre-Post position: {of, in}, Coordinating Conj.: {and, or, but} and punctuation: {. :}.

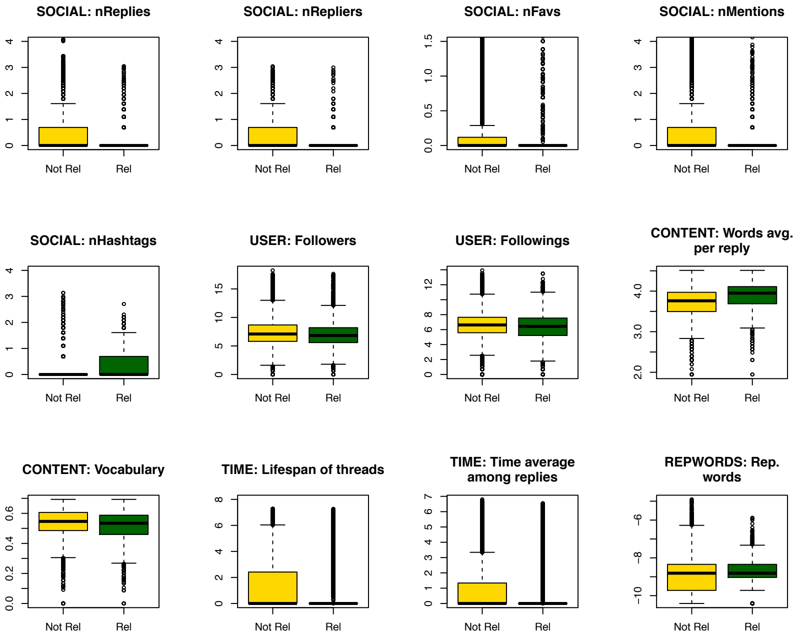


Figure 8.2: Single features of threads that contribute most to thread relevance.

Building?", we found some tweets of threads that contain the following sentences:

```
- How many floors are there in the Empire State Building? #EmpireStateBuilding
   #NYCtrivia #NYC #iloveNY
- How many floors does the Empire State Building have? http://t.co/9GNrVWRjXl
   #uselessfacts, #didyouknow
```

Notice that the meaning of hashtags are about trivia (the first tweet) and to spread though the social network (the second tweet).

In addition, we found that long threads of factoid QA are more likely to be non-relevant (number of replies of Figure 8.2) since factoid QA questions tend to generate low numbers of replies. When this number is high, conversation threads are related to jokes or other comments between repliers. Similar behavior occurs with @mentions.

On the other hand, we found that relevant threads have a low number of replies but each reply has high number of words (Figure 8.2). This means that short replies (with concrete answers) are not necessary relevant threads. With these results, we can support the findings of Lin et al. [69] who said that users prefer answers in a paragraph rather than the exact phrase.

With respect to time, large differences between replies of a thread indicate that probably the thread is non-relevant . We propose to retrieve more of these kinds of features in a future work in order to study the influence of time in some time-sensitive questions, for example, to try to resolve questions quickly in a critical event such as an earthquake or terrorist attack.

Word Embeddings in microblogs QA. Another important finding of our study has to do with the best way to use word embeddings for LTR in QA. Molino et al. [78] use this feature as our D_WEMB, i.e., calculating the distance (or similarity) between the query and potential answers based on the skip-gram representation. While they had good results with this feature of "distributional semantics", it was ranked only 30th among other text quality metrics. In our case, we used distances, but also the word embedding representation directly as features, which yielded excellent results, ranking as the 2nd most important feature set. This indicates that for microblog QA it is better to use the values of the embedding dimensions as features rather than a single value which aggregates them.

Transfer Learning in microblog QA. Our manual inspection of results indicates that transfer learning can be a potential way to perform non-factoid QA, by using a model pre-trained for factoid QA. However, for future work we want to generalize this result and study special features of non-factoid tasks. Moreover, for this experiment we might need to collect a larger non-factoid ground truth dataset.

Final Comments. We studied several sets of features (extracted from threads) using ranking methods. We performed a quantitative evaluation on a factoid QA dataset and a informative evaluation with non-factoid questions. In particular, we found that part-of-speech

(POS) features are key for determining thread relevance. However, these features require other sets of features such as social and content to improve performance. This is also something that has been observed in other cQA platforms, in comparison with other platform specific text-quality features, where POS features provide better indication for finding the best answers than other features related to network behavior and user profiles [78].

Regarding the LTR ranking framework, MART consistently outperforms the other methods, and the best results are obtained when combining several sets of features. We propose to explore LTR based on deep learning in the future.

For the non-factoid analysis, we get good MRR using the same model trained for factoid QA (transfer learning). The good performance is because the model learned to rank answers based on factoid QA and, non-factoid answers (related to recommendations) have similar characteristics. This means, for example, that POS features perform well because nouns take relevance in both cases (factoid and non-factoid about recommendations). Finally, in a more extensive and manual experiment, we found out that removing tweets that are not part of a thread can improve the MRR. We acknowledge some loss of information, but in the analysis we demonstrate that the trade off is worth the loss. In fact, using this enhancement, we could not find answers in just 2 of 40 non-factoid questions (5%).

Overall, the evidence obtained in our current work provides a positive answer for our first and third research questions (**RQ1** and **RQ3**) indicating that Twitter historical data can in fact be used to answer questions, including complex questions related to recommendations. In a future work we propose to incorporate more complex questions such as context-aware (temporal and geo-spatial) as well as personalized questions. Regarding **RQ2**, our results show that content quality features such as POS play an important role for ranking, even more than Social and User type features. This is also something that has been observed in other cQA platforms, in relation to other platform specific text-quality features that also provided better indication for finding best answers than other features related to network behavior and user profiles [78].

Chapter 9

Conclusion

In this thesis we investigated the feasibility of conducting QA using microblog data. In particular, we studied whether microblogs such as Twitter can be leveraged to obtain answers for information needs paraphrased as questions.

We created a ranking model using well-formed factoid questions and then applied it to non-factoid questions related to recommendations. Our results validate the potential for using microblog data for factoid and non-factoid QA, identifying the most informative features as well as the best LTR model.

The evidence obtained in our current work provides a positive answer for **RQ1** and **RQ3**, indicating that Twitter's historical data can in fact be used to answer questions, including complex questions.

Summarizing, the mayor findings of this study are:

- Twitter has relevant information for answering questions (**RQ1**).
- We demonstrate that conversation threads provide more relevant information than individual tweets for QA (**RQ3**).
- POS (part-of-speech) features are key to determining the thread relevance. However, the ranking improves when we add others such as social and content features (**RQ2**). In the same way, SOCIAL features are discriminative for non-relevant prediction (except hashtags).
- Through a transfer learning technique, our best ranking model using factoid QA was useful in answering non-factoid questions related to recommendations (**RQ3**).

Due to the similarity between microblogs (streaming platforms), we consider that these results can be reproducible in other platforms of this nature.

Limitations and future guidelines. The main limitation found during the factoid QA experiment was that our approach could only retrieve answers for about 40% of the questions. This issue, in particular, could be mitigated by improving the coverage provided by the different query formulations. However, part of this issue is also due to the timeliness of the information poured by users into microblogs, which does not always match the queries in the standard factoid datasets.

We also note that our initial factoid dataset, based on TREC challenges (between 1999 and 2005), does not have topics related to current events, which are much more likely to be discussed in Twitter [62]. The most recent factoid dataset that we used corresponds to the year 2005; however the oldest tweet that we were able to retrieve was from 2007. This time gap between our ground-truth questions and our candidate answers can very likely explain why we were unable to find matching tweets for an important number of questions. In general, when examining questions that did not retrieve any candidate answers, we observed that they corresponded to dated topics. Nevertheless, we believe that candidate answer recall could be improved if we used a more complete Twitter dataset, as opposed to a small sample of the data stream as we do now. In this same line of research, a periodic collection of questions asked by users in social media and their corresponding candidate answers could contribute to creating an up-to-date knowledge base of timely topics.

In addition, the proposed model for factoid questions works with non-factoid questions, but it needs changes. Non-factoid questions in Twitter present more noise than factoid questions. For future studies we propose to modify (or add) Query Formulations (QF) for this specific task. In addition, we found that threads without replies can be eliminated to improve ranking performance (MMR).

Regarding factoid evaluation, our ground truths did not contain current events, which are the most commented on Twitter topics. This most likely explains why we only have a 34% overlap between the results provided by Twitter and the questions in our TREC data. In this context, finding a 34% intersection could be interpreted as the probability of finding information about any random anachronic topic on Twitter at any given time. Of course, we would need more experiments to validate this issue.

We expect to conduct a larger evaluation on other types of non-factoid questions, perform an in-depth analysis on the effect of certain attributes, study other features, add more query formulations (based on question type) for non-factoid questions and improve our dataset though a crowd sourcing task to generate a real ranking of threads. Furthermore, we will add other distributional models and neural architectures such as RNN and LSTM to enhance our framework [47, 60, 118].

We are aware that our approach does not consider whether or not some thread has the correct answer or not. We only explore the features that allow us to understand whether an answer is in a thread or not. For example, for the factoid question "*How did Bob Marley die?*", we found a thread with these four replies:

```
R1: Cancer I Think
R2: Marijuana overdose.
R3: No such thing.
R4: He got shot.
```

Even though our ranking model put this thread in the top-10 position, the approach is not capable of determining correct and incorrect answers. In this case, only the reply R1 is correct. We will address this issue in a future work as well.

Moreover, we detect another relevant issue. We realized that around 50% of our questions were related to places or present some spatial reference. For example, "*Can anyone recommend good accommodation for Paris?*" or "*Anyone wanna recommend any bars in downtown Phoenix*?" Hence, another future approach would be to personalize the answers based on a spatial context.

Finally, in future studies we expect to address time-sensitive questions in critical events. The main goal of this study would be to analyze the first questions that arise in emergency situations, and devise a way of answering them quickly.

Bibliography

[1] Arvind Agarwal, Hema Raghavan, Karthik Subbian, Prem Melville, Richard D Lawrence, David C Gondek, and James Fan. Learning to Rank for Robust Question Answering. In *Proceedings of CIKM 2012*, number 2, page 833, New York, USA, 2012. IBM Thomas J. Watson Research Center, ACM.

[2] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilard Gilad Mishne. Finding High-quality Content in Social Media. In *Proceedings of WSDM 2008*, pages 183–193. Emory University, ACM, 2008.

[3] Hadi Amiri, Zheng-Jun Zha, and Tat-Seng Chua. A Pattern Matching Based Model for Implicit Opinion Question Identificacion. In *Proceedings of AAAI 2013*, pages 46–52, 2013.

[4] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to Compose Neural Networks for Question Answering. In *Proceedings of NAACL-HLT 2016*, volume cs.CL, pages 1545–1554, San Diego, CA, USA., 2016.

[5] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *Proceedings of ICCV 2015*, pages 2425–2433, Santiago, Chile, 2015. IEEE Computer Society.

[6] Muhammad Asaduzzaman, Ahmed Shah Mashiyat, Chanchal K Roy, and Kevin A Schneider. Answering questions about unanswered questions of stack overflow. In *Proceedings of MSR 2013*, pages 97–100. ˜IEEE Press, may 2013.

[7] Sitaram Asur and Bernardo A. Huberman. Predicting the future with social media. In *Proceedings of WI-IAT 2010*, pages 492–499, 2010.

[8] Lars Backstrom, Jon Kleinberg, Lillian Lee, and Cristian Danescu-Niculescu-Mizil. Characterizing and Curating Conversation Threads: Expansion, Focus, Volume, Reentry. In *Proceedings of WSDM 2013*, pages 13–22, 2013.

[9] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*, volume 463. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, jan 1999.

[10] Jiang Bian, Yandong Liu, Eugene Agichtein, and Hongyuan Zha. Finding the Right Facts in the Crowd: Factoid Question Answering Over Social Media. In *Proceeding of*

WWW 2008, page 467, 2008.

[11] David M Blei. Probabilistic Topic Models. *Communications of the ACM*, 55(4):77–84, 2012.

[12] Mohan John Blooma and Jayan Chirayath Kurian. Clustering Similar Questions In Social Question Answering Services. In *Proceedings of PACIS 2012*, pages 1–15, 2012.

[13] Mohamed Bouguessa, Benoît Beno\^\it Dumoulin, and Shengrui Wang. Identifying Authoritative Actors in Question-Answering Forums: the Case of Yahoo! Answers. In *Proceedings of KDD 2008*. University of Sherbrooke, ACM, 2008.

[14] Danah Boyd, Scott Golder, and Gilad Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *Proceedings of HICSS 2010*, pages 1–10, Washington, DC, USA, 2010. IEEE.

[15] Wladmir C. Brandão, Rodrygo L. T. Santos, Nivio Ziviani, Edleno S. de Moura, and Altigran S. da Silva. Learning to expand queries using entities. *Journal of the Association for Information Science and Technology*, 65(9):1870–1883, sep 2014.

[16] Christopher J C Burges. From RankNet to LambdaRank to LambdaMART: An Overview. Technical report, jun 2010.

[17] Christopher J C Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to Rank using Gradient Descent. In *Proceedings of ML 2005*, pages 89–96, New York, New York, USA, 2005. ACM Press.

[18] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of ICML 2007*, pages 129–136, New York, New York, USA, 2007. Microsoft Research Asia, ACM.

[19] Jilin Chen, Rowan Nairn, and Ed Chi. Speak little and well: recommending conversations in online social streams. In *Proceedings of the CHI 2011*, page 217, New York, New York, USA, may 2011. Palo Alto Research Center, ACM.

[20] Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the CHI 2010*, pages 1185–1194, New York, New York, USA, apr 2010. Massachusetts Institute of Technology, ACM.

[21] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. Collaborative personalized tweet recommendation. In *Proceedings of SIGIR 2012*, page 661, New York, New York, USA, 2012. ˜ACM Request Permissions.

[22] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P Kuksa. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[23] Gao Cong, Long Wang, Young-In Song, and Yueheng Sun. Finding Question-Answer

Pairs From Online Forums. In *Proceedings of SIGIR 2008*, page 467, New York, New York, USA, 2008. ˜ACM Request Permissions.

[24] David Cossock and Tong Zhang. Subset Ranking Using Regression. *Proceedings of the 19th Annual Conference on Learning Theory*, 4005:605–619, 2006.

[25] Koby Crammer and Yoram Singer. Pranking With Ranking. *Neural Information Processing Systems (NIPS)*, pages 641–647, 2001.

[26] Daniel Hasan Dalip, Marcos Andre Gonçalves, Marco Cristo, and Pavel Calado. Exploiting User Feedback to Learn to Rank Answers in Q&A Forums: a Case Study with Stack Overflow. In *Proceedings of SIGIR 2013*, pages 543–552, Dublin, Ireland, 2013.

[27] Ernesto Diaz-Aviles, Lucas Drumond, Lars Schmidt-Thieme, and Wolfgang Nejdl. Real-time top-n recommendation in social streams. In *Proceedings of RecSys 2012*, page 59, New York, New York, USA, 2012. ACM Press.

[28] Simon Dooms, Toon De Pessemier, and Luc Martens. MovieTweetings: a Movie Rating Dataset Collected From Twitter C3Places-Using ICT for co-creation of inclusive public places View project DECODE: High-Resolution Lead Design for Closed Loop Deep Brain Stimulation View project MovieTweetings: a Movie Rating. In *In Proceeding of Workshop on Crowdsourcing and Human Computation for Recommender Systems RecSys 2013*.

[29] Gideon Dror, Dan Pelleg, Oleg Rokhlenko, and Idan Szpektor. Churn prediction in new users of Yahoo! answers. In *Proceedings of WWW 2012*, number January 2016, page 829, New York, New York, USA, 2012. ACM Request Permissions.

[30] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An Empirical Study on Learning to Rank of Tweets. In *Proceedings of COLING 2010*, pages 295–303, Beijing, China, 2010. Microsoft Research Asia, Association for Computational Linguistics.

[31] Bradley Efron and Robert J Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, Berlin Heidelberg, 1994.

[32] Miles Efron and Megan Winget. Questions are content: A taxonomy of questions in a microblogging environment. In *Proceedings of ASIST 2010*, volume 47. American Society for Information Science, 2010.

[33] Lei Fang, Minlie Huang, and Xiaoyan Zhu. Question Routing in Community Based QA: Incorporating Answer Quality and Answer Content. In *Proceedings of MDS 2012*, pages 1–8, New York, New York, USA, aug 2012. ˜ACM Request Permissions.

[34] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An Efficient Boosting Algorithm for Combining Preferences. *Journal of machine Learning Research*, 4:933–969, 2003.

[35] Yoav Freund and Robert E Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[36] Jerome H Friedman. Greedy Function Approximation: a Gradient Boosting Machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[37] Mengyi Gao and Xiang Zhang. A Movie Recommender System from Tweets Data. Technical report.

[38] Saptarshi Ghosh, Naveen Sharma, Fabricio Benevenuto, Niloy Ganguly, and Krishna Gummadi. Cognos: Crowdsourcing Search for Topic Experts in Microblogs. In *Proceedings of SIGIR 2012*, page 575, 2012.

[39] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das 0001, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, Noah A Smith, Brendan O&apos;Connor, Dipanjan Das 0001, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, Noah A Smith, and Dipanjan Das. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of ACL 2011*, number 2, pages 42–47. Cornell University Library, 2011.

[40] Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. Multimedia Lab @ ACL W-NUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations. In *Procceeding of ACL 2015*, pages 146–153, 2015.

[41] Jinwen Guo, Shengliang Xu, Shenghua Bao, and Yong Yu. Tapping on the potential of Q&A community by recommending answer providers. In *Proceedings of CIKM 2008*, page 921, New York, New York, USA, 2008. ACM Press.

[42] Zoltan Gyongyi, Georgia Koutrika, Jan Pedersen, and Hector Garcia-Molina. Questioning Yahoo! Answers. In *Proceedings of WWW 2008*, 2008.

[43] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers*, pages 115–132, 2000.

[44] Jose Herrera, Barbara Poblete, and Denis Parra. Retrieving Relevant Conversations for Q&A on Twitter. In *Proceeding of SIGIR 2015 (Workshop of SPS)*, volume 1421, pages 21–25, Santiago, Chile, 2015.

[45] Jose Herrera, Barbara Poblete, and Denis Parra. Learning to Leverage Microblog Data for QA. In *Proceeding of ECIR 2018*, pages 507–520, Grenoble, France, 2018.

[46] Andrew Hickl. Answering Questions with Authority Categories and Subject Descriptors. In *Proceeding of CIKM 2008*, pages 1261–1270, 2008.

[47] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[48] C Honey and S C Herring. Beyond Microblogging: Conversation and Collaboration via Twitter. In *Proceedings of HICSS 2009*, pages 1–10, Big Island, HI, USA, 2009. ˜IEEE Computer Society.

[49] Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. A Neural Network for Factoid Question Answering over Paragraphs. *Proceedings of EMNLP 2014*, pages 633–644, 2014.

[50] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why We Twitter: Understanding Microblogging Usage and Communities. In *Proceedings of KDD (Workshop WebKDD) 2007*, pages 56–65, New York, New York, USA, aug 2007. ˜ACM Request Permissions.

[51] Jiwoon Jeon, W Bruce Croft, and Joon Ho Lee. Finding Similar Questions in Large Question and Answer Archives. In *Proceedings of CIKM 2005*, page 84, New York, New York, USA, 2005. ˜ACM Request Permissions.

[52] Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. A Framework to Predict the Quality of Answers with Non-Textual Features. In *Proceedings of SIGIR 2006*, page 228, New York, New York, USA, 2006. ACM.

[53] Daniel Jurafsky and James H Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. In *Speech and Language Processing*, volume 21, pages 0–934. Prentice Hall, Pearson Education International, New Jersey, USA, 2009.

[54] Pawel Jurczyk and Eugene Agichtein. Discovering authorities in question answer communities by using link analysis. In *Proceedings of CIKM 2007*, pages 919–922, New York, New York, USA, 2007. ˜ACM Request Permissions.

[55] Pawel Jurczyk and Eugene Agichtein. Hits on Question Answer Portals: Exploration of Link Analysis for Author Ranking. In *Proceedings of SIGIR 2007*, page 845, New York, New York, USA, 2007. ACM Request Permissions.

[56] Jie Kang, Kyle Condiff, Shuo Chang, Joseph A. Konstan, Loren Terveen, and F. Maxwell Harper. Understanding How People Use Natural Language to Ask for Recommendations. In *Proceedings of RecSys 2017*, RecSys '17, pages 229–237, New York, NY, USA, 2017. ACM.

[57] Jong-hoon Oh Kentaro and Torisawa Chikara. Why Question Answering using Sentiment Analysis and Word Classes. In *Proceeding of EMNLPCN 2012*, number July, pages 368–378, 2012.

[58] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[59] Oleksandr Kolomiyets and Marie-Francine Francine Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, dec 2011.

[60] Mandy Korpusik, Shigeyuki Sakaki, Francine Chen, and Yan-Ying Chen. Recurrent Neural Networks for Customer Purchase Prediction on Twitter. In *Proceeding of RecSys 2016*, 2016.

[61] Onur Kucuktunc, Ingmar Weber, B. Barla Cambazoglu, Hakan Ferhatosmanoglu, Ingmar Weber, Hakan Ferhatosmanoglu, B. Barla Cambazoglu, Hakan Ferhatosmanoglu, Ingmar Weber, and Hakan Ferhatosmanoglu. A Large-Scale Sentiment Analysis for Yahoo! Answers. In *Proceedings of WSDM 2012*, pages 633–642, New York, New York, USA, feb 2012. ˜ACM Request Permissions.

[62] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *Proceedings of WWW 2010*, page 591, New York, New York, USA, 2010. ACM Press.

[63] Won-Jo Lee, Kyo-Joong Oh, Chae-Gyun Lim, and Ho-Jin Choi. User Profile Extraction From Twitter for Personalized News Recommendation. In *Proceedings of ICACT 2014*, pages 779–783. Global IT Research Institute (GIRI), 2014.

[64] Baichuan Li, Tan Jin, Michael R Mr Lyu, Irwin King, and Barley Mak. Analyzing and predicting question quality in community question answering services. In *Proceedings of WWW 2012*, pages 775–782, New York, New York, USA, 2012. ˜ACM Request Permissions.

[65] Baichuan Li and Irwin King. Routing Questions to Appropriate Answerers in Community Question Answering Services. In *Proceedings of CIKM 2010*, page 1585, New York, New York, USA, 2010. Chinese University of Hong Kong, ACM.

[66] Baichuan Li, Irwin King, and Michael R. Lyu. Question Routing in Community Question Answering: Putting Category in Its Place. In *Proceedings of CIKM 2011*, page 2041, New York, New York, USA, oct 2011. ˜ACM Request Permissions.

[67] Baichuan Li, Xiance Si, Michael R. Lyu, Irwin King, and Edward Y. Chang. Question Identification on Twitter. In *Proceedings of CIKM 2011*, pages 2477–2480, New York, New York, USA, 2011. ACM Request Permissions.

[68] Ping Li, Christopher Burges, and Qiang Wu. McRank: Learning to Rank UsingMultiple Classification and Gradient Boosting. *Advances in Neural Information Processing Systems 20*, 7:845–852, 2008.

[69] Jimmy Lin, Dennis Quan, Vineet Sinha, Karun Bakshi, David Huynh, Boris Katz, and David R. Karger. What Makes a Good Answers? The Role of Context in Question Answering Systems. In *Proceedings of INTERACT 2003*, number September, page 1006, 2003.

[70] Tie-Yan Liu. *Learning to rank for information retrieval.* Springer Science & Business Media, Berlin Heidelberg, 2011.

[71] Y Liu, T Alexandrova, and T Nakajima. Using Stranger as Sensors: Temporal and Geo-Sensitive Question Answering via Social Media. In *Proceedings of WWW 2013*,

2013.

[72] Zhe Liu and Bernard J Jansen. Factors Influencing the Response Rate in Social Question and Answering Behavior. In *Proceedings of CSCW 2013*, page 1263, New York, New York, USA, 2013. ACM Request Permissions.

[73] Zhe Liu and Bernard J. Jansen. Predicting potential responders in social Q&A based on non-QA features. In *Proceedings of CHI 2014*, pages 2131–2136, New York, New York, USA, 2014. ACM Press.

[74] Zhe Liu and Bernard J. Jansen. A Taxonomy for Classifying Questions Asked in Social Question and Answering. In *Proceedings of CHI EA 2015*, pages 1947–1952, New York, New York, USA, 2015. ACM Press.

[75] Matteo Magnani, Danilo Montesi, and Luca Rossi. Conversation retrieval for microblogging sites. *Information Retrieval*, 15(3-4):354–372, 2012.

[76] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.

[77] Tomas Mikolov, Chenm Kai, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *NIPS*, cs.CL:1–12, 2013.

[78] Piero Molino, Luca Maria Aiello, and Pasquale Lops. Social Question Answering: Textual, User, and Network Features for Best Answer Prediction. *ACM Transactions on Information Systems*, 35(1):4–40, 2016.

[79] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. A Comparison of Information Seeking Using Search Engines and Social Networks. In *Proceedings of ICWSM 2010*, pages 23–26, 2010.

[80] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. What do people ask their social networks, and why? In *Proceedings of CHI 2010*, page 1739, New York, New York, USA, 2010. ACM Press.

[81] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. Improved Part-of-Speech Tagging for Online Conversational Text With Word Clusters. In *Proceedings of ACL 2013*, number June, pages 380–390, Atlanta, Georgia, USA AD -, 2013. Cornell University Library.

[82] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. *Stanford InfoLab*, 54(1999-66):1–17, 1998.

[83] Aditya Pal and Joseph A Konstan. Expert Identification in Community Question Answering: Exploring Question Selection Bias. In *Proceedings of CIKM 2010*, pages 1505–1508, New York, New York, USA, 2010. ACM.

[84] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions*

on Knowledge and Data Engineering, 22:1345—-1359, 2010.

[85] Weike Pan, Erheng Zhong, and Qiang Yang. Transfer Learning for Text Mining. In *Mining Text Data*, number January, pages 223–257. Springer US, Boston, MA, jan 2012.

[86] Denis Parra and Shaghayegh Sahebi. *Recommender systems: Sources of knowledge and evaluation metrics.* 2013.

[87] Dan Pelleg, Oleg Rokhlenko, Mark Shovman, Idan Szpektor, and Eugene Agichtein. The Crowd is Not Enough: Improving User Engagement and Satisfaction Through Automatic Quality Evaluation. In *Proceeding of SIGIR 2015 (Industrial Track)*, pages 1–10, 2015.

[88] Marco Pennacchiotti and Ana-Maria Popescu. Democrats, Republicans and Starbucks Afficionados: User Classification in Twitter. In *Proceedings of KDD 2011*, page 430, New York, New York, USA, 2011. ACM Press.

[89] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using Twitter to recommend real-time topical news. In *Proceedings of RecSys 2009*, pages 385–388, New York, New York, USA, 2009. ACM Request Permissions.

[90] Mingcheng Qu, Guang Qiu, Xiaofei He, Cheng Zhang, Hao Wu, Jiajun Bu, and Chun Chen. Probabilistic Question Recommendation for Question Answering Communities. In *Proceedings of WWW 2009*, volume 12, page 1229, New York, New York, USA, 2009. ˜ACM.

[91] Daphne Ruth Raban. Self-presentation and the value of information in Q&A websites. *Journal of the American Society for Information Science and Technology*, 60(12):2465–2473, 2009.

[92] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. Technical report, 2018.

[93] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. Technical report, 2016.

[94] Luis R Sánchez. *Repurchase Intention for Lodging Recommendation.* Phd thesis, 2017.

[95] Johannes Schantl, Rene Kaiser, Claudia Wagner, and Markus Strohmaier. The utility of social and topical factors in anticipating repliers in Twitter conversations. In *Proceedings of WebSci 2013*, pages 376–385, New York, New York, USA, may 2013. Graz University of Technology, ACM Press.

[96] Aliaksei Severyn and Alessandro Moschitti. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *Proceedings of SIGIR 2015*, pages 373–382, 2015.

[97] Chirag Shah and Jefferey Pomerantz. Evaluating and predicting answer quality in

community QA. In *Proceeding of SIGIR 2010*, page 411, New York, New York, USA, 2010. Rutgers University, ACM Press.

[98] Paul Sharoda, Lichan Hong, and Ed H Chi. Is Twitter a Good Place for Asking Questions? A Characterization Study. In *Proceedings of ICWSM 2011*, pages 1–4, Barcelona, Spain, 2011.

[99] Anna Shtok, Gideon Dror, Yoelle Maarek, and Idan Szpektor. Learning From the Past: Answering New Questions with Past Answers. In *Proceedings WWW 2012*, page 759, New York, New York, USA, 2012. ACM.

[100] Rosaria Silipo, Iris Adae, Aaron Hart, and Michael Berthold. Seven Techniques for Dimensionality Reduction. *KNIME.com*, 2014.

[101] Swapna Somasundaran. QA with Attitude : Exploiting Opinion Type Analysis for Improving Question Answering in On-line Discussions and the News. In *Proceeding of ICWSM 2007*, page cited 45, 2007.

[102] Daniel Sousa, Lu\'\is Luís Lu\'\is Sarmento, and Eduarda Mendes Rodrigues. Characterization of the Twitter @replies network: are user ties social or topical? In *Proceedings of SMUC 2010*, page 63, New York, New York, USA, 2010. University of Porto, ACM.

[103] Cleyton Souza, Jonathas Magalhaes, Evandro Costa, Joseana Joseama Fechine, Jonathas Magalhães, Evandro Costa, and Joseana Joseama Fechine. Routing questions in Twitter: An effective way to qualify peer helpers. *Proceedings of WI 2013*, 01:109–114, 2013.

[104] Ke Sun, Yunbo Cao, Xinying Song, Young-In Song, Xiaolong Wang, and Chin-Yew Lin. Learning to Recommend Questions Based on User Ratings. In *Proceeding of CIKM 2009*, pages 751–758, New York, New York, USA, 2009. ACM.

[105] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to Rank Answers on Large Online QA Collections. In *Proceedings of ACL 2008*, number June, pages 719–727, Columbus, Ohio, USA, 2008.

[106] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to Rank Answers to Non-Factoid Questions from Web Collections. *Computational Linguistics*, 37(2):351–383, 2011.

[107] Matthew E Taylor and Peter Stone. Transfer Learning for Reinforcement Learning Domains: A Survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.

[108] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology.*, 61(12):2544–2558, 2010.

[109] E.M. Voorhees. The Philosophy of Information Retrieval Evaluation. *Lecture Notes in Computer Science (CLEF 2001)*, pages 355–370, 2001.

[110] Baoxun Wang, Bingquan Liu, Xiaolong Wang, Chengjie Sun, and Deyuan Zhang. Deep Learning Approaches to Semantic Relevance Modeling for Chinese Question-Answer Pairs. *ACM Transactions on Asian Language Information Processing*, 10(4):1–16, dec 2011.

[111] Jianshu Weng, Ee Peng Lim, Jing Jiang, and Qi He. TwitterRank: Finding Topic-Sensitive Influential Twitterers. In *Proceedings of WSDM 2010*, page 261, New York, New York, USA, feb 2010. Pennsylvania State University, ACM.

[112] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise Approach to Learning to Rank: Theory and Algorithm. In *Proceedings of ICML 2008*, pages 1192–1199, New York, New York, USA, jul 2008. Chinese Academy of Sciences, ACM.

[113] Jun Xu and Hang Li. AdaRank: a Boosting Algorithm for Information Retrieval. In *Proceedings of SIGIR 2007*, pages 391–398, New York, New York, USA, 2007. Microsoft Research Asia, ACM.

[114] Yuto Yamaguchi, Tsubasa Takahashi, Toshiyuki Amagasa, and Hiroyuki Kitagawa. TURank: Twitter user ranking based on user-tweet graph analysis. In *Lecture Notes in Computer Science*, volume 6488 LNCS, pages 240–253, Berlin, Heidelberg, 2010. University of Tsukuba, Springer Berlin Heidelberg.

[115] Jiang Yang, Meredith Ringel Morris, Jaime Teevan, Lada A Adamic, and Mark S Ackerman. Culture Matters: A Survey Study of Social Q&A Behavior. In *Proceeding of AAAI 2011*, pages 409–416, jul 2011.

[116] Liu Yang, Qingyao Ai, Damiano Spina, Ruey-Cheng Chen, Liang Pang, W. Bruce Croft, Jiafeng Guo, and Falk Scholer. Beyond Factoid QA: Effective Methods for Non-factoid Answer Sentence Retrieval. *Lecture Notes in Computer Science (ECIR 2016).*, 9626:115–128, 2016.

[117] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked Attention Networks for Image Question Answering. In *Proceedings of CVPR 2016*, pages 21–29, Las Vegas, Nevada, USA, 2015.

[118] Borui Ye, Guangyu Feng, Anqi Cui, and Ming Li. Learning Question Similarity with Recurrent Neural Networks. In *Proceeding of ICBK 2017*, pages 111–118. IEEE, aug 2017.

[119] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. Deep Learning for Answer Sentence Selection. *NIPS deep learning workshop*, cs.CL:9, 2014.

[120] Hamed Zamani, Azadeh Shakery, and Pooya Moradi. Regression and Learning to Rank Aggregation for User Engagement Evaluation. In *Proceedings of RecSys (RecSys Challenge) 2014*, pages 29–34, New York, New York, USA, 2014. ACM Press.

[121] Zhe Zhao and Qiaozhu Mei. Questions About Questions: an Empirical Analysis of Information Needs on Twitter. In *Proceedings of WWW 2013*, pages 1545–1556, New York, New York, USA, may 2013. University Michigan Ann Arbor, International World

Wide Web Conferences Steering Committee.

[122] X. Zheng, Z. Hu, A. Xu, D. R Chen, K. Liu, and B. Li. Algorithm for Recommending Answer Providers in Community-Based Question Answering. *Journal of Information Science*, 38(1):3–14, 2012.

[123] Joey Tianyi Zhou, Sinno Jialin Pan, Ivor W Tsang, and Shen-shyang Ho. Transfer Learning for Cross-Language Text Categorization through Active Correspondences Construction. In *Proceedings of AAAI 2016*, number Settles, pages 2400–2406, 2016.

[124] Tom Chao Zhou, Michael R. Lyu, and Irwin King. A Classification-Based Approach to Question Routing in Community Question Answering. In *Proceeding of WWW 2012*, page 783, New York, New York, USA, 2012. ACM.

Appendices

A1. Characterization of Recommendation Tweets

We evaluated non-factoid questions related to recommendations, because these are the questions most asked in social networks [80]. During the inspection of these types of questions on Twitter, we found several recommendation tweets, which are not necessarily paraphrased as questions. Therefore, we extended our study to provide an initial exploratory analysis of recommendation tweets to determine how users ask for recommendations in microblogs.

We define a **recommendation tweet** as *"an explicit recommendation provided by a user with a positive or negative preference about something or someone"*. We retrieve these kinds of tweets and study the types of recommendations that users give in microblogs and how users request and provide recommendations. In the future, we will create a taxonomy of recommendation tweets (based on similar studies by Efron and Winget [32] and Liu and Jansen [74]). Figure 1 provides examples of explicit recommendations in different topics; movies, music, books, etc.

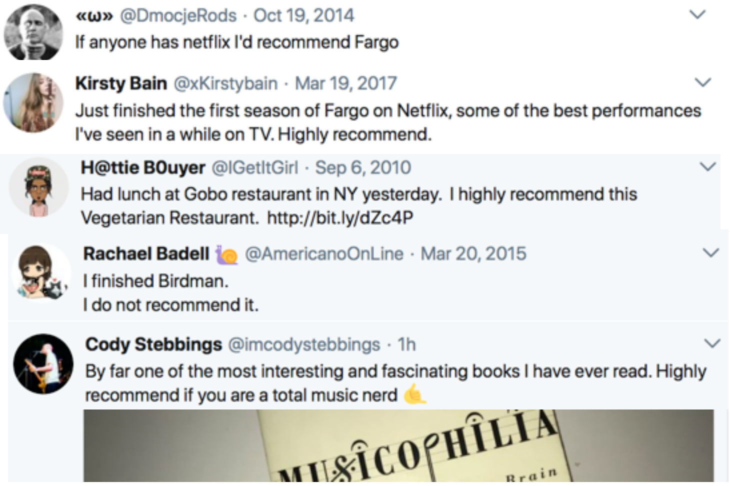
Request and Provide Recommendations. In a initial inspection, we found that users provide and request recommendations in two ways: (1) a user **asks** for a recommendation. Figure 2 shows an example of recommendation thread. (2) a user **provides** a recommendation without any previous request as shown in Figure 1.

We addressed (1) in Chapter 7 by inspecting non-factoid questions related to recommendations. The tweets of (2) are interesting because users are not forced to provide them; they do it on their own initiative. The fact that the user opens the social network, writes a recommendation and shares it could be more valuable than a recommendation posted in a specialized website [80].

Although there are related studies that personalize recommendations based on user tweets [21, 37, 63, 89], there are no studies that retrieve and identify these kinds of messages. To identify them, we analyze the content of tweets to find special characteristics. We find that some are occurrences of phrases such as *"I recommend"*, *"I highly recommend"*, *"I strongly recommended"*, *"we recommend"*, etc. Moreover, there is another relevant component; users use different tones for recommendations. For example, the first two tweets in Figure 1 refer to the TV series called *Fargo*. The second one seems to be stronger than the first with special phrases as "*...some of the best performances...*" or "*I highly recommend*". This demonstrates a measure of strength between recommendation tweets.

As mentioned, there are no studies or tools that detect recommendation tweets. The identification of this information can be useful for other related tasks. For example, we could compute a score based on Twitter recommendations on a specific item and then compare with other specialized platforms such as IMDB[1]. In fact, there is a study of Dooms et al. [28] (MovieTweetings) which captures tweets generated by IMDB. These tweets are generated, not by users directly, but by automatically pressing a button at IMDB's webpage. The value of our work, differing from the latter work, is that tweets (threads) are written directly by

[1] IMDB is one of the most popular movie databases which includes reviews and ratings.



«ω» @DmocjeRods · Oct 19, 2014
If anyone has netflix I'd recommend Fargo

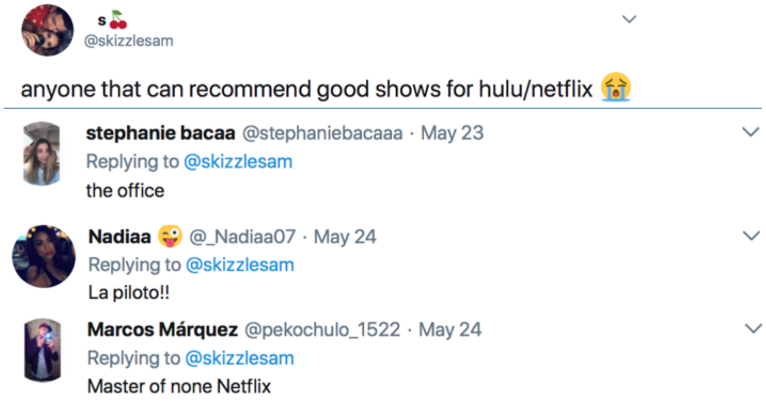
Kirsty Bain @xKirstybain · Mar 19, 2017
Just finished the first season of Fargo on Netflix, some of the best performances I've seen in a while on TV. Highly recommend.

H@ttie B0uyer @IGetItGirl · Sep 6, 2010
Had lunch at Gobo restaurant in NY yesterday. I highly recommend this Vegetarian Restaurant. http://bit.ly/dZc4P

Rachael Badell 🐌 @AmericanoOnLine · Mar 20, 2015
I finished Birdman.
I do not recommend it.

Cody Stebbings @imcodystebbings · 1h
By far one of the most interesting and fascinating books I have ever read. Highly recommend if you are a total music nerd 🤙

Figure 1: Recommendation tweets taken from Twitter. Users provide recommendations without any request.



s 🍒
@skizzlesam

anyone that can recommend good shows for hulu/netflix 😭

stephanie bacaa @stephaniebacaaa · May 23
Replying to @skizzlesam
the office

Nadiaa 😜 @_Nadiaa07 · May 24
Replying to @skizzlesam
La piloto!!

Marcos Márquez @pekochulo_1522 · May 24
Replying to @skizzlesam
Master of none Netflix

Figure 2: An example of how users ask for a recommendation followed by three replies (answers). In this case, the initial tweet is not paraphrased as a question.

users. Therefore, we could provide a complete and extensive dataset of recommendations based on Twitter. This could be available to the community for future research purposes. Moreover, identifying recommendation tweets can encourage other studies such as, the tweets in a premiere of a movie, predicting box office success, among others. Asur et al. [7] conducted a similar study predicting box office movie success. However, they consider only the frequency of tweets related to the movie, but there are other relevant elements to consider such as time, content and geo-spatial features (among others).

Retrieval. We collected 400 recommendation tweets using the Twitter API between May 28^{th}, 2018 between 20:26 UTC and 20:41 UTC. We retrieve 200 tweets using the query *recommend* and 200 with *recommended*. Table 1 features the description of this dataset. We then manually classified each tweet in one of these four categories:

- **Books**: includes articles and comic recommendations.
- **Movies**: includes TV and TV series recommendations.
- **Other**: places, recommends following other users, games, other resources (url), technologies, factual knowledge, etc.

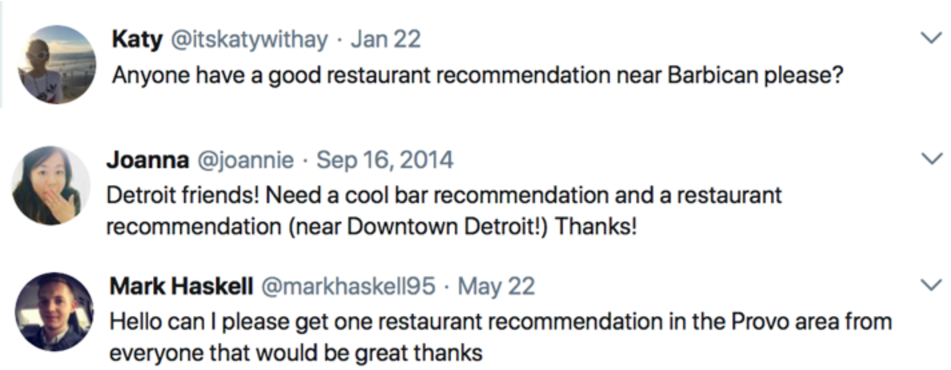
We discard tweets presenting these characteristics:

- tweets that are difficult to understand: "*And I have to say that every bottle of wine recommended here was amazing.*"
- tweets related to following someone: "*We don't recommend literally following someone.*"
- tweets that are not clear because they are headed to a specific user (a reply) or belong to a conversation: "*@KreekCraft or maybe a canon or gopro but i still recommend dslr*" or "*@drjompatterson Thank you Jo! Highly recommended MSc module!*"
- repeated tweets (related to marketing) "*Sleepy is a Korean rapper. He is in fact the rapper that recommended RM to a Big Hit producer he was friends with...*".

Type	**Value**
Initial retrieve	400
Recommendation questions (request)	76 (20,0%)
Post-filtering	117 (29,3%)
# users that recommend more than once	2
Recommendation Topics:	
Books	38 (32,5%)
Movies	15 (12,8%)
Music	8 (6,8%)
Other	56 (47,9%)

Table 1: Dataset description of recommendation tweets.

Final Comments. In the analysis, we realize that the Books category presents the largest number of tweets. However, it is probably that recommendation tweets are influenced by the date/time. It means that at the time that we retrieved the tweets, the Books category



Katy @itskatywithay · Jan 22
Anyone have a good restaurant recommendation near Barbican please?

Joanna @joannie · Sep 16, 2014
Detroit friends! Need a cool bar recommendation and a restaurant recommendation (near Downtown Detroit!) Thanks!

Mark Haskell @markhaskell95 · May 22
Hello can I please get one restaurant recommendation in the Provo area from everyone that would be great thanks

Figure 3: Users asking for recommendations based on a certain location.

presents more tweets, but in the night of the same day, it is possible that the Movies category presents a high frequency. The Movies category could be affected by time and date, because it depends on when a movie or TV series premiers in each country. For example, the premieres in Chile are on Thursdays, hence we could expect more frequency of tweets in Chile related to Movies on Thursdays.

In summary, there are several challenges that arise from this preliminary study. One difficulty is to determine the entities from recommendation, such as the name of a book, movie, restaurants, etc. Although there are tools to extract entities from text such as NER[2], sometimes the movie's names are not easy to detect with these tools. For instance, "*Recommend me some tv series or movies to watch friends.*" or "*Recommend a funny uncomplicated series friends*". We do not know whether the following tweets are about the TV series called *Friends* or if the word Friends is just a noun. Likely, some topic modeling such as LDA [11] can be helpful as well.

Regarding this preliminary experiment, it was useful to encourage further research in this area. Although we retrieved tweets related to recommendations, we have not considered adding Recommender System's methods [86]. In the future, we can extend this study and focus on user preferences. For example, if we know the user's preference of some specific topic, we could find a user's expertise in some topics, such as movies, or readers. Moreover, if we know the movie genre preference of users, we could recommend similar movies. In fact, we found in our data two users that recommend more than once in the same topic (Table 1). In short, the next step of our research can be to add some Recommender System's methods in order to recommend pieces of information relevant for specific users.

Finally, we could anticipate recommendation requests using the context such as the geospatial information and time. For example, Figure 3 shows users asking about different restaurants for recommendations in a specific area. In these cases, the requests have a context related to place, date and time. Although Twitter is suitable for asking time sensitive questions, it is not easy to identify the context of the content. In fact, it is an unsolved problem in the QA research community.

We expect to address all of these interesting issues in future work.

[2] Named Entities Recognition: https://nlp.stanford.edu/software/CRF-NER.html