Extracting Keywords from Multi-party Live Chats

Su Nam Kim
School of Information Technology
Monash University
Clayton, VIC, Australia
su.kim@monash.edu.au

Timothy Baldwin
Dept. of Computing and Information Systems
The University of Melbourne
VIC, Australia
tb@ldwin.net

Abstract

Live chats have become a popular form of communication, connecting people all over the globe. We believe that one of the simplest approaches for providing topic information to users joining a chat is keywords. In this paper, we present a method to automatically extract contextually relevant keywords for multi-party live chats. In our work, we identify keywords that are associated with specific dialogue acts as well as the occurrences of keywords across the entire conversation. In this way, we are able to identify distinguishing features of the chat based on structural information derived from live chats and predicted dialogue acts. In evaluation, we find that using structural information and predicted dialogue acts performs well, and that conventional methods do not work well over live chats.

1 Introduction

Keywords or **keyphrases**[1] are an effective way of representing the core topic of a document, and can effectively summarize and/or help index documents. They are usually found in the form of either simplex nouns (e.g. *library*) or noun phrases (e.g. *social issue*). They have been studied in the past to provide topic-related information for many applications such as text summarizers, search engines and indexers. For example, Barzilay and Elhadad (1997) used keywords as semantic meta-information for summarizers. DÁvanzo and Magnini (2005) used them to organize documents for search engines. Dredze et al. (2008) used keywords as summaries of email in order to better manage and prioritize emails. Hammouda et al. (2005) used keywords extracted from multiple documents in order to discover the topics of documents for clustering. Gutwin et al. (1999) used automatically extracted keywords to refine the queries to improve precision of search in an online library browser.

There has been much research on automatic keyword extraction (Frank et al., 1999; Turney, 1999; Hulth, 2003, inter alia). The majority of work has been done over specific domains such as scientific articles and newspapers, including the recent SemEval-2010 shared task on keyword extraction (Kim et al., 2010b). A small minority of researchers have used different sources of data such as email (Dredze et al., 2008) and HTML documents (Mori et al., 2004), as outlined in Section 2. However, existing approaches tend not to work well when applied to different target sources, and are often susceptible to domain-specific features of the target documents (e.g. structure).

In this paper, our aim is to automatically extract keywords for multi-party live chats. Live chats are essentially text-based dialogues, with less disfluencies than spoken dialogues but greater scope for overlapping utterances and out-of-sequence subthreading (Ivanovic, 2008). Researchers have variously proposed to use *dialogue acts* (or *DAs*) to analyze the structure of discourses. In this paper, we are primarily interested in extracting keywords, but hypothesise that keywords not only serve as summaries of live chats, but they can also track the top-

[1] In this work, we use the term *keywords* for consistency, while noting that it can be used to refer to multiword terms.

26th Pacific Asia Conference on Language,Information and Computation pages 199–208

ics of the conversation. Furthermore, keywords provided at different points of a chat can benefit participants who are absent from the chat for a period of time. This would be especially beneficial to multi-party conversations which pose great challenges due to tangled and asynchronous nature of the interaction. One may easily imagine that keywords could provide contextualizing information for a conversation, helping a new participant to join a conversation mid-stream. Hence, the ability to extract keywords at any given time during the conversation has the potential to enhance the user-friendliness of live chat systems.

In this research, we target multi-party written dialogues for keyword extraction due to their popularity on the web, the ability for participants to readily join and leave chats, and the novel semi-asynchronous nature of interactions. However, we believe that the proposed methodology could be adapted to spoken dialogues, noting the challenges of automatic speech recognition, and the import of acoustic and prosodic features in keyword extraction.

In analyzing chat data, we observed that keywords vary over time due to topic changes as the conversation progresses. Also, we found that keywords are highly associated with specific dialogue acts. As such, we explored the structural information and dialogue acts predicted by our dialogue act classification system to accommodate the characteristics of live chats. During evaluation, we compared our proposed methods with the well-known KEA keyword extraction system. For our work, we collected data from live chat forums from the US Library of Congress (see Section 3 for details). Unlike casual chats (e.g. NPS live chats), the conversations are based on specific issues, and are thus similar to task-oriented settings such as meetings.

2 Related Work

Keyword (or keyphrase) extraction has been studied over the years, with the primary aim of deducing the topic of a document. The task involves selecting keyword candidates, ranking candidates in terms of the relatedness to the document topic(s), and evaluating the system and/or looking for suitable learning methods. A major portion of prior research work has focused on the ranking problem and has mostly used statistical approaches with various sets of features from symbolic resources and linguistically-motivated heuristics and machine learners (Frank et al., 1999; Turney, 1999; Hulth, 2003; Nguyen and Kan, 2007; Kim et al., 2010b). Since our effort focuses on feature engineering for live chats, we detail the previous efforts on feature engineering and variety of datasets keyword extraction has been applied to.

KEA (Frank et al., 1999) was one of the earliest keyword extraction systems, and was based on TF·IDF and the location of first appearance of each term in the document. Hereafter, we will refer to this term as *first appearance*. The GenEx system (Turney, 1999) employed nine heuristic features based exclusively on morphosyntax, such as word length and phrase frequency. Hulth (2003) used TF·IDF, first appearance and keyphraseness[2] as the basis of his method, and added POS tags assigned to candidate terms based on the observation that POS patterns such as (NN NN) and (JJ NN) are more frequent among keywords. Nguyen and Kan (2007) extracted keywords using structural information such as the document title and section headings derived from scientific articles. Wan and Xiao (2008) used a document clustering method to extract salient words, then utilized those to rank the candidates. Liu et al. (2009) developed an unsupervised method using TF·IDF and variants thereof. The main approach is to cluster the terms with respect to the sub-topics, rank candidates in each cluster, then select top-ranked candidates as keywords. Li et al. (2010) proposed a method based on semantic similarity among n-ary phrases, based on Wikipedia entities and links, and used the weighted Girvan-Newman algorithm for candidate ranking. More recently, Kim et al. (2010b) proposed a keyword extraction shared task over scientific articles. Participants used a broad range of features based on document structure, semantic similarity and various document and term heuristics.

Keyword extraction has also been carried out on various types of documents. Scientific articles and news articles are often the target of keyword extraction (Hulth, 2003; Nguyen and Kan,

[2]The intuition is that what is a good keyword in one context is likely to be a good keyword in similar contexts.

2007; Medelyan, 2009; Kim et al., 2010b). Hulth (2003) extracted 2,000 abstracts of journal articles from Inspec that contained controlled and uncontrolled terms assigned by professional indexers. Nguyen and Kan (2007) collected a dataset containing 120 computer science articles and labeled them with both author- and reader-assigned keywords. Medelyan (2009) collected 180 full-text publications from CiteULike using user tags. More recently, the SemEval-2010 keyword extraction task used 100 and 144 scientific articles with author and reader-assigned keywords for testing and training, respectively. In the biomedical domain, Schutz (2008) obtained 1,323 files with gold-standard answers and predictions from PubMed. Wan and Xiao (2008) developed a set of 308 documents with up to 10 manually-assigned keywords using newswire documents from DUC 2001. Dredze et al. (2008) used keywords as summaries of email in order to better manage and prioritize emails.

3 Dialogue Acts for Multi-party Live Chats

While developing keyword data for live chats, we observed a strong correlation between dialogue acts and keywords. As such, we chose to first annotate chat data with dialogue acts. We collected multi-party live chat data from forums from the US Library of Congress. The live chats contain 33 online discussions that the Library's Educational Outreach team hosted for teachers between 2002 and 2006.

To define dialogue acts that suit this data, we investigated existing sets of dialogue acts from both spoken dialogues and live chats. Many can be found in both spoken and written dialogues based on the Dialogue Act Markup in Several Layers (DAMSL) scheme (Allen and Core, 1997). For live chats, Wu et al. (2002) and Forsyth (2007) defined 15 dialogue acts for casual online conversations based on previous sets (Samuel et al., 1998; Shriberg et al., 1998; Jurafsky et al., 1998; Stolcke et al., 2000). Ivanovic (2008) proposed 12 dialogue acts applying DAMSL for customer service chats.

Given the fact that our live chat forum data is closer to customer service chats in terms of the nature of the data (e.g. question, request, gratitude etc.), we decided to adopt the set from Ivanovic (2008) and added two more dialogue acts – BACKGROUND and OTHER. The list of dialogue acts and examples can be found in Table 1.

We selected 15 forums containing at least 200 utterances. The data was first segmented into discourse units, and sentence tokenized. Then, we cleaned the data by tokenizing emoticons/expletives (e.g. : −), *wow*), email addresses (e.g. *learningpage@loc.gov*), URLs (e.g. *http://memory.loc.gov*), locations (e.g. *Texas*), and institutes (e.g. *University of Houston*) into *EMOTION, EMAIL, URL, LOCATION, INSTITUTE*, respectively. We also replaced user names with *USER_ID* to anonymize the data. Our final dataset contains 5,276 utterances from 15 live chat forums, after removing system log data.[3] The proportion of instances corresponding to each dialogue act is shown in Table 1.

We annotated the dialogue acts in order to analyze the distribution of keywords over different dialogue acts, and further, to use dialogue acts as features for the keyword extractor. To manually assign dialogue acts, we used two annotators including the first author. The annotators have past experience in conducting annotations for similar tasks. Before the actual annotation task, we also did a pilot test using live chat forums which were not selected for our final dataset. The initial agreement was 81.4% and kappa value was 0.74, indicating a well-defined annotation task. Note that we employed an automatic dialogue act classifier based on previous work (Forsyth, 2007; Kim et al., 2010a) to pre-assign and post-edit dialogue acts for the keyword extraction task. Using the system significantly reduced annotation effort, and yet was found to not bias the annotation process, based on small-scale experimentation with and without the DA predictions. Details of the DA prediction model are provided in Section 4.

4 Dialogue Act Prediction

In this section, we present our attempt to automatically extract dialogue acts in order to use them for our main task: keyword extraction. It is important to note that we employ previously-proposed features without modification. Our goal in using these meth-

[3] System logs indicate the status of participants, such as a participant joining or departing

Dialog Act	Example	Percent	Dialog Act	Example	Percent
OPENING	Hi, Greeting!	3.03	RESPONSE-ACK	yes, great, i agree,..	11.73
CLOSING	bye, good night,..	1.55	WH-QUESTION	What is this?	3.26
BACKGROUND	i am user2, i teach 4th grad	4.76	YN-QUESTION	is there a website for .. ?	5.84
THANKING	thanks, thank you for ..	6.54	YES-ANSWER	yes, sure,	1.67
EXPRESSION	: −), wow, oh!	7.71	NO-ANSWER	no, nope	0.28
STATEMENT	we have a website for photo gallery.	47.76	DOWNPLAY	no problem, you're welcome!	0.49
REQUEST	click this, go to..	4.97	OTHER	or, but	0.40

Table 1: Dialogue act tagset: definitions and examples

ods is to avoid manual annotation on dialogue acts, and thus we do not detail the effectiveness of previous methods nor evaluate our system against previous methods.

We explored various features from recent work (Forsyth, 2007; Kim et al., 2010a) to automatically predict dialogue acts. Our features are based on high-frequency terms with respect to dialogue acts from Forsyth (2007), and contextual, structural, and dialogue act interaction from Kim et al. (2010a). Note that in Forsyth (2007), the author used the term *keyword*. Keywords in Forsyth (2007) are defined as terms which are frequently associated with specific dialogue acts, and thus differ from our definition of keywords in this work. We thus refer to Forsyth's "keywords" as high-frequency terms. Finally, we developed a linear-chain conditional random field-based dialogue act classification system using Mallet (McCallum, 2002),[4] based on Kim et al. (2010a). We used 15 fold-cross validation (i.e. one dialogue for test and remainings for training), as our data contains 15 live chats.

After experimenting with various features, we found that contextual and high-frequency terms w.r.t. dialogue act features generally performed well, while structural and dialogue act interaction features did not achieve high accuracy, despite claims to the contrary in other studies. We hypothesise that since our data contains large numbers of users (unlike the two-party chat data of Ivanovic (2008), e.g.), the resulting entanglement of sub-threads confuses the dialogue act tagger. To elaborate, stemming tended to reduce errors caused by ill-formed words (e.g. *noooooo* as *no*). *High-frequency terms* also performed well since they are highly associated with specific dialogue acts (e.g. *hi, hello* for OPENING, *ok, great* for RESPONSE-ACK). However, it takes intensive manual intervention to extract such words associated with particular dialogue acts. Also, *user information* from structural features improves performance. We observed that user names mentioned in the dialogues resolve the entanglement to some degree, and thus perform well for dialogue act classification (e.g. *you're right, USER25!!*). The features used to automatically predict dialogue acts are listed below:

- *Stemmed Bag-of-Words*
- *Highly frequent terms* per dialogue act
- *User/Participant information*

To summarize, our best dialogue act classifier achieved an accuracy of 82.79%. We postulate that the lower accuracy compared to that reported in previous work (e.g. Forsyth (2007; Kim et al. (2010a)) was mainly due to the different nature of the chats as well as the higher number of participants. However, we found this was sufficient to semi-automate the annotation process.

5 Feature Engineering

To build the baseline system, we first used three features from KEA: (1) *TF·IDF*, one of most frequently used features, measures the relateness between the document topic(s) and candidate terms; (2) *first appearance* is a heuristic that indicates the locality of the keywords; that is, keywords often appear at the

[4] http://mallet.cs.umass.edu

beginning or end as well as specific parts of a document (e.g. Frank et al. (1999; Nguyen and Kan (2007)); and (3) *keyphraseness*, based on the observation that keywords tend to share across documents with the same or similar topics.

For our system, we developed new features based on observation, and structural information. First, we observed that keywords occur across chats since the discussed topics change across time, unlike the globally-relevant keywords typically found in documents such as scientific articles and news articles. Ideally, a topic shift detection method could identify boundaries of topic change. However, automatic topic detection would introduce errors and manual topic detection would involve high cost and time. Thus, we leave this issue for our future work. Finally, we decided to equally split each live chat into 10 smaller documents and to treat each as a single smaller document to compute IDF. To compensate for the erroneous topic boundaries due to the equal split, we used a variant of the sliding window approach. That is, we also include the last 10% of dialogues from the previous split document, resulting in each document partition containing approximately 11% of the whole document.

Secondly, we found that some dialogue acts (e.g. STATEMENT, REQUEST) tend to contain most of the keywords. Also, utterances made by host users tend to have more keywords than those by non-host users. Based on these, we introduced two features: (1) TF of keywords in utterances tagged with selected dialogue acts; and (2) TF of keywords in utterances made by host users. Statistical analysis of these observations is provided in Section 6.

Thirdly, we used the distribution of candidate keywords over the 10 sub-documents. Ideally, when documents are well split by sub-topics, keywords would appear in only a few sub-documents and not the whole document.

We summarize our tested features below:

- Baseline Features from `KEA`

  F1: TF·IDF$_{all}$ IDF over all documents

  F2: First Appearance

  F3: Keyphraseness

- Structural and Dialogue Features

  F4: TF·IDF$_{split}$ IDF over $\frac{1}{10}$ splits of the document

  F5: TF over utterances tagged with selected dialogue acts The association between keywords and utterances tagged with selected dialogue acts, in the form of raw, local proportion, and global proportion

  F6: TF over Host Utterances The association between keywords and utterances made by host users, in the form of raw, local proportion, and global proportion

  F7: TF over 10 Sub-documents Distribution of TF over each 10% of the original document, in the form of the raw count, local proportion, and global proportion. The distribution of TF is represented in 10 vectors, each representing 10% of the original document.

For features F5, F6 and F7, we tested three different ways of calculating the feature values. *Raw* is the raw term count. *Local* is computed using the proportion of term counts in selected utterances against that in all utterances; the motivation behind this is that instead of using raw counts, we check if the term occurrence in selected utterances has an impact. Finally, *global proportion* is computed using the term frequency in selected vs. all utterances, and is a combination of raw and local proportion values.

1. *raw*: TF in utterances tagged with selected dialogue acts only (*selU*); cf. TF in all utterances is marked as *allU*.

2. *local proportion*: $\frac{\text{TF}\in\text{selU}}{\text{TF}\in\text{allU}}$

3. *global proportion*: $\frac{\frac{\text{TF}\in\text{selU}}{|selU|}}{\frac{\text{TF}\in\text{allU}}{|allU|}}$

6 Data

To evaluate our proposed keyword extraction method, we collected keywords from 15 live chat forums. To simplify the task, we only allowed the annotators to extract simplex nouns as keywords.[5] One annotator manually extracted keywords, then

[5] During the pilot annotation test, we observed that the vast majority of keywords are simplex nouns.

the second (and more experienced for this task) annotator reviewed the extracted keywords. For disagreed keywords, two annotators met to finalize the keywords.

In total, 148 keywords were assigned to the 15 live chats. We checked the occurrence of keywords over 14 dialogue acts in manually-labeled dialogues and found that all keywords were found in one of four dialogue acts — STATEMENT, REQUEST, YN-QUESTION, WH-QUESTION — which make up 61.73% of utterances in our data. Table 2 shows the distribution of keywords over the 14 dialogue acts.

We also observed that 140 keywords are found in utterances made by host users (94.59% coverage), which make up 52.50% of utterances in the data. As candidates, we used lemmatized nouns with frequency ≥ 2 after removing stop words, and the EMOTION, URL, EMAIL, INSTITUTE and LOCATION tokens. After selecting the keyword candidates, we checked the coverage of keywords in the candidates. Across all utterances, we extracted 1,717 token candidates including 144 keyword types. On the other hand, in utterances tagged with one of the 4 selected dialogue acts, we got 1,494 token candidates, making up 142 keyword types. It shows that using only the 4 selected dialogue acts reduced the token candidate set by 12.99% but missed only 2 keyword types. This underlines the strong association between keywords and the selected dialogue acts.

7 Evaluation

7.1 Experimental Setup

In the preprocessing step, we performed POS tagging with `Lingua::EN::Tagger`, lemmatization with `morph` (Minnen et al., 2001) and stemming with `English Porter stemmer`.[6]

To build the automatic keyword extractor, we used naive Bayes to rank the keyword candidates with various features, following Kim et al. (2010b).[7] Likewise, to run the system, we used 15 fold-cross validation since we have 15 live chats. Note that when computing the counts of term frequencies for features F5, F6, and F7, we used the training data to avoid overfitting. For evaluation, we used the evaluation metric used in Kim et al. (2010b) but changed the top-N selection to use the top-5, 7 and 10 ranked candidates, since the average number of keywords per document is 9.9.

7.2 Results

Tables 3 and 4 show the performance (micro-averaged precision, $\mathcal{P}_\mu$, recall, $\mathcal{R}_\mu$ and F-score, $\mathcal{F}_\mu$) over 3 different settings of top-N candidates. We also present the performance using all utterances (marked as **allU**) vs. only those utterances corresponding to one of the four dialogue acts which our dialogue act classifier automatically labeled (marked as **selU**). In addition, we used two different sets of documents — original documents vs. split documents — in order to compute TF·IDF. As a result, we have four sets of experiments for baseline features — (Original Documents vs. Split Documents for IDF) $\times$ (All Utterances vs. Selected Utterances)

For features F5 $\sim$ F7, since we already observed better performance with F4 (TF·IDF over split documents), we test these features with F4 only.

While the dialogue act tagger was used to semi-automate the DA annotation, it is important to note that the dialogue act labels used in this experiment are those taken directly from the automatic DA tagger. Our baseline system, `KEA`, was also tested over all utterances as well as selected utterances only.

Overall, the systems performed better when using TF·IDF over split documents for both all utterances and selected utterances. In our description of the occurrence of keywords in dialogues, we observed that using smaller document chunks would contain keywords, as the conversation has a specific topic to discuss in each time frame. Even with the original `KEA` using all three features (i.e. F1–F3), using TF·IDF alone performed much better. We observed that the first-occurrence heuristic (which indicates term locality) does not effectively identify keywords in live chat data, since the documents themselves have sequential structure and likewise, keywords occur all across the documents. This shows that keywords in dialogues are more associated with time than document structure, as is the case with scientific and/or news articles. As for the reappearance of keywords

[6] Using the Perl implementation available at `http://tartarus.org/~martin/PorterStemmer/`

[7] We also experimented with a maximum entropy learner, but found the results to be near-identical, and omit them from this paper.

DA	keyword	DA	keyword	DA	keyword
STATEMENT	1127	WH-QUESTION	62	THANKING	17
REQUEST	119	RESPONSE-ACK	37	OPENING	13
YN-QUESTION	99	BACKGROUND	31	CLOSING	1

Table 2: Distribution of keywords over dialogue acts

Feature	Top 5			Top 7			Top 10		
	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$
Baseline Features using Original Documents (KEA)									
F1	42.67	21.62	28.70	39.05	27.70	32.41	24.67	25.00	24.83
F1+F2	16.00	8.11	10.76	18.10	12.84	15.02	9.33	9.46	9.39
F1+F3	32.00	16.22	21.53	31.43	22.30	26.09	18.00	18.24	18.12
F1+F2+F3†	16.00	8.11	10.76	18.10	12.84	*15.02*	9.33	9.46	9.39
Baseline Features using Split Documents									
F4	53.33	27.03	35.88	57.14	40.54	**47.43**	33.33	33.78	33.55
F4+F2	16.00	8.11	10.76	20.00	14.19	16.60	10.00	10.14	10.07
F4+F3	8.00	4.05	5.38	18.10	12.84	15.02	4.67	4.73	4.70
F4+F2+F3	16.00	8.11	10.76	20.00	14.19	16.60	10.00	10.14	10.07
Dialogue Features using Split Documents									
F4+F5$_{raw}$	48.00	24.32	32.28	54.29	38.51	45.06	30.00	30.41	30.20
F4+F5$_{tf}$	1.33	0.68	0.90	3.81	2.70	3.16	2.00	2.03	2.01
F4+F5$_{percent}$	1.33	0.68	0.90	3.81	2.70	3.16	2.00	2.03	2.01
F4+F6$_{raw}$	49.33	25.00	33.18	54.29	38.51	45.06	30.00	30.41	30.20
F4+F6$_{tf}$	5.33	2.70	3.58	7.62	5.41	6.33	4.00	4.05	4.02
F4+F6$_{percent}$	5.33	2.70	3.58	7.62	5.41	6.33	4.00	4.05	4.02
F4+F7$_{raw}$	48.00	24.32	32.28	55.24	39.19	45.85	29.33	29.73	29.53
F4+F7$_{tf}$	12.00	6.08	8.07	10.48	7.43	8.70	6.00	6.08	6.04
F4+F7$_{percent}$	12.00	6.08	8.07	11.43	8.11	9.49	6.00	6.08	6.04

Table 3: Effectiveness of keyword extraction over **All Utterances (allU)** (the baseline [KEA] is marked with †, and its performance is in italics; the best performance is bold-faced). **Original Documents** means IDF computed as is, while **Split Documents** means IDF calculated over $\frac{1}{10}$ splits of the document.

(i.e. seen keyword heuristics), it did not work well since we have only 15 chats and many keywords occurred in most of the chats (whether as keywords or not).

Comparing all utterances vs. selected utterances, the performance was very similar. However, in some cases, using selected utterances performed better (e.g. with $F4 + F6$, 45.06% and 51.38% for allU and selU, respectively). We estimate that since the discarded utterances are relatively short and often contain general terms, even if we include these utterances, the effect of these discarded utterances is insignificant. However, given the best performance over the two different utterance sets, we can argue that using selected utterances achieves higher performance with much fewer candidates.

Among Top-5, 7, and 10, surprisingly, we found that the performance with the top-7 rated candidates consistently exceeded that with the top-10 rated candidates. To analyze this, we checked $\mathcal{P}_\mu$, $\mathcal{R}_\mu$ and $\mathcal{F}_\mu$, and found that precision tends to drop as we add more candidates.

Finally, we observed that our novel features based on dialogue structure and dialogue acts (F5–F7) contributed to correctly extract keywords, especially over the top-5 candidates. We found that the utterance author information (F6) is particularly effective at identifying keywords with high accuracy. Simi-

Feature	Top 5			Top 7			Top 10		
	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$
Baseline Features using Original Documents (KEA)									
F1	41.33	20.95	27.81	40.95	29.05	33.99	24.67	25.00	24.83
F1+F2	18.67	9.46	12.56	23.81	16.89	19.76	12.00	12.16	12.08
F1+F3	32.00	16.22	21.53	31.43	22.30	26.09	17.33	17.57	17.45
F1+F2+F3†	18.67	9.46	12.56	23.81	16.89	*19.76*	12.00	12.16	12.08
Baseline Features using Split Documents									
F4	53.33	27.03	35.88	58.10	41.22	48.23	33.33	33.78	33.55
F4+F2	20.00	10.14	13.46	24.76	17.57	20.55	12.67	12.84	12.75
F4+F3	5.33	2.70	3.58	16.19	11.49	13.44	5.33	5.41	5.37
F4+F2+F3	20.00	10.14	13.46	24.76	17.57	20.55	12.67	12.84	12.75
Dialogue Features using Split Documents									
F4+F5$_{raw}$	48.00	24.32	32.28	51.43	36.49	42.69	30.00	30.41	30.20
F4+F5$_{tf}$	1.33	0.68	0.90	2.86	2.03	2.37	0.67	0.68	0.67
F4+F5$_{percent}$	1.33	0.68	0.90	2.86	2.03	2.37	0.67	0.68	0.67
F4+F6$_{raw}$	53.33	27.03	35.88	61.90	43.92	**51.38**	32.67	33.11	32.89
F4+F6$_{tf}$	6.67	3.38	4.49	9.52	6.76	7.91	4.67	4.73	4.70
F4+F6$_{percent}$	6.67	3.38	4.49	9.52	6.76	7.91	4.67	4.73	4.70
F4+F7$_{raw}$	42.67	21.62	28.70	53.33	37.84	44.27	26.00	26.35	26.17
F4+F7$_{tf}$	13.33	6.76	8.97	14.29	10.14	11.86	8.00	8.11	8.05
F4+F7$_{percent}$	12.00	6.08	8.07	11.43	8.11	9.49	6.67	6.76	6.71

Table 4: Effectiveness of keyword extraction over **Selected Utterances (selU)** (the baseline [KEA] is marked with †, and its performance is in italics; the best performance is bold-faced). bold-faced). **Original Documents** means IDF computed as is, while **Split Documents** means IDF calculated over $\frac{1}{10}$ splits of the document.

larly, since keywords tend to appear in selected dialogue acts, term frequency over the utterances labeled with these dialogue acts only produced good results compared to term frequency over all utterances. Likewise, the distribution of keywords over the 10 sub-documents (F7) contributed to higher performance compared to the baseline system. Among the three different values we tested, we found that using raw counts performed the best. We speculate that due to the small size of the data, the normalised values did not work well.

8 Conclusion

In this paper, we proposed the task of automatic keyword extraction problem over multi-party live chats in order to provide in situ topic information. Based on our observations, we developed a system using structural information and automatically predicted dialogue acts, and achieved preliminary results applying an existing dialogue act classification method to live chats. Unlike previous research (e.g. Forsyth (2007; Kim et al. (2010a)), features based on structure and interaction did not perform well since multi-party live chats impose problems due to the tangled and asynchronous nature of chats. Finally, we showed that our method achieved higher performance than KEA, which implies that conventional methods (KEA in this paper) do not work well over structured data like live chats and web forums.

In this research, we found structural features to be one of the most important features in correctly identifying keywords. To date, none of topic detection methods appear to work. On the other hand, detecting topic boundaries with higher accuracy would improve the performance of the keyword extractor. As such, we leave this for future work.

References

James Allen and Mark Core. 1997. Draft of damsl: Dialog act markup in several layers.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL/EACL 1997 Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

Ernesto DÁvanzo and Bernado Magnini. 2005. A keyphrase-based approach to summarization:the lake system. In *Proceedings of Document Understanding Conferences*, pages 6–8.

Mark Dredze, Hanna M. Wallach, Danny Puller, and Fernando Pereira. 2008. Generating summary keywords for emails using topics. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 199–206.

Eric N. Forsyth. 2007. Improving automated lexical and discourse analysis of online chat dialog. Master's thesis, Naval Postgraduate School.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-manning. 1999. Domain specific keyphrase extraction. In *Proceedings of the 16th International Joint Conference on AI*, pages 668–673.

Carl Gutwin, Gordon Paynter, Ian Witten, Craig Nevill-Manning, and Eibe Frank. 1999. Improving browsing in digital libraries with keyphrase indexes. *Journal of Decision Support Systems*, 27:81–104.

Khaled M. Hammouda, Diego N. Matute, and Mohamed S. Kamel. 2005. Corephrase: keyphrase extraction for document clustering. In *Proceedings of MLDM*, pages 265–274.

Annette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223.

Edward Ivanovic. 2008. Automatic instant messaging dialogue using statistical models and dialogue acts. Master's thesis, the University of Melbourne.

Daniel Jurafsky, Elizabeth Shriberg, Barbara Fox, and Traci Curl. 1998. Lexical, prosodic, and syntactic cues for dialog acts. In *Proceedings of ACL/COLING-98 Workshop on Discourse Relations and Discourse Markers*, pages 114–120.

Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 2010a. Classifying dialogue acts in 1-to-1 live chats. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 862–871.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010b. SemEval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Uppsala, Sweden.

Decong Li, Sujian Li, and Wenjie Li. 2010. A semi-supervised key phrase extraction approach: learning from title phrases through a document semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 296–300.

Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 620–628.

A. K. McCallum. 2002. Mallet: A machine learning for language toolkit.

Olena Medelyan. 2009. *Human-competitive automatic topic indexing*. Ph.D. thesis, University of Waikato.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Junichiro Mori, Yutaka Matsuo, Mitsuru Ishizuka, and Boi Faltings. 2004. Keyword extraction from the web for personal metadata annotation. In *4th International Workshop on Knowledge Markup and Semantic Annotation*, pages 51–60.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Key phrase extraction in scientific publications. In *Proceeding of International Conference on Asian Digital Libraries*, pages 317–326.

Ken Samuel, Carbeery Sandra Carberry, and K. Vijay-Shanker. 1998. Dialogue act tagging with transformation-based learning. In *Proceedings of COLING/ACL 1998*, pages 1150–1156.

Alexander Thorsten Schutz. 2008. Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods. Master's thesis, National University of Ireland.

Elizabeth Shriberg, Rebecca Bates, Paul Taylor, Andreas Stolcke, Daniel Jurafsky, Klaus Ries, Noah Coccaro, Rachel Martin, Marie Meteer, and Carol Van Ess-Dykema. 1998. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41(3-4):439–487.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Peter Turney. 1999. Learning to extract keyphrases from text.

Xiaojun Wan and Jianguo Xiao. 2008. Collabrank: Towards a collaborative approach to single-document

keyphrase extraction. In *Proceedings of 22nd International Conference on Computational Linguistics*, pages 969–976, Manchester, UK.

Tianhao Wu, Faisal M. Khan, Todd A. Fisher, Lori A. Shuler, and William M. Pottenger. 2002. Posting act tagging using transformation-based learning. In *Proceedings of the Workshop on Foundations of Data Mining and Discovery, IEEE International Conference on Data Mining*.