***Abstract:***

News agents publishing so many online news stories for readers but only some of those stories get attention and be popular. Social media shares of online new stories may give opinions about popularity of new story. We are trying the find out that if the new article in Milliyet.com.tr will be shared in Ekşi Sözlük or not.

***Introduction :***

As growing social media, blogosphere, streaming sites, new types of data become

available that can be mined for valuable knowledge. For example online

chatter can be used to predict sales ranks of books.

Many of posts on in social media is a response to news events, investigating events and their impact as reflected in social media has become an important for media analysts.

Purpose of this work is predicting the popularity before publishing it. Predicting popularity might help in determining the desirability of an article. Also if news agents knew which articles can be popular, they can spend their sources for popular ones. Also they can offer articles which readers might be more interested. If news agents can gain more readers, they can earn more money from ad givers.

Our aim in this paper is to predict popularity of news articles before publication. By popularity, we mean if the article will be shared on Ekşi Sözlük which is the most popular social media organization in Turkey, which is Turkish origin.

This work makes several contributions. First, it explores the dynamics of stories in Milliyet.com.tr. Second, it introduces the problem of predicting the popularity of a news article. Third, it provides a set of textual and semantic features that can be used to predict the share in Ekşi Sözlük of a news story before publication. Fourth, it provides an evaluation of the introduced features. Fifth an error analysis identifies possible causes for classification failure.

***RELATED WORK***

Most of the popularity prediction works based on early comments/likes/retweets. So that means they need content to be published. In this experiment, we are trying to find out the popularity before publication.

Article “Predicting the Volume of Comments on Online News Stories”[1] using 5 features set for finding popularity; Surface,Cummulative, Textual, Semantic and Real World. Surface feature looking for is article has image in it or how many authors contributed to that article. Cummalive feature looking for how many articles published in the same hour. Real Word Feature looking for if wheather is good or bad. So we used only Textual and Semantic features because those are the only features with pure text mining.

***EXPLORING NEWS***

Our data consits on 449 news stories from Milliyet.com.tr. 60 of them is not shared in Ekşi Sözlük, others are shared on Ekşi Sözlük between February 2016 to August 2016. February to July used for training and August stories are used for test.

***Gathering data; !!!! YİĞİT !!!!***

We turn to our first research question: What are the dynamics of user generated comments on news articles? Accoring to article “Predicting the Volume of Comments on Online News Stories”[1] , there are 5 features that effect on the popularity; surface,cumulative, textual, semantic, real world. Only textual and semantic features are about text mining, so that we used those 2 feature in our work.

Textual feature means get term frequency of top-100 terms ranked by their log-likelihood score for all stories.

Semantic feature means get person count mentioned in article, location count mentioned in article and organization count mentioned in article.

***FEATURE ENGINEERING***

For textual feature, we first found most common 200 terms in all stories. We skiped stop words in turkish, and used Resha Turkish Stemmer[2] for stemming operations. As we predict, “Başbakan”, “Cumhurbaşkanı”, “Futbol”, “Polis”, “Asker”, “Şehit”, “Patlama”, “Terör” were some of the most common terms, those were the most popular topics in 2016’s Turkey.

Articles which include more popular terms may be more popular. So we measured term frequencies of all articles, which were in most common 200 terms.

For semantic feature , we used İTÜ Natural Language Processing Tool[3]. With this tool, we were able to find Named Entities of each word in the articles. According to pipeline we used below schema.

Named Entity Recognizer

Morphological Disambiguator

Morphological Analyzer

Thus we were able to get person, organization and location counts for each news article.

After that we used Random Forest Algorithm[4]. We had 4 numerical data(Term Frequency of popular terms, Person Count, Organization Count, Location Count) and 1 label(Popular or Not).

We re-implement ironmanMA’s algorithm for our purpose.

***Experimental set-up***

Ekşi Sözlük does not provide api, so we manually found news stories shared in Ekşi Sözlük .

After that we extract news stories to folders for each month.

Then we found 200 most common terms and wrote to MostCommonTerms file.

Then we processed input file for textual features and wrote to TermFrequencies file.

Since İTÜ NLP service goes offline most of the time, we used threads for each month, and wrote to output file SemanticVariables .

With using 3 output files, we created TreeData Files for each test and training stories and wrote to TreeData file and TestTreeData file.

Using tree data files we used random forest implemantation for results. Then we calculated F1 scores and accuracies.

RESULTS

….

ERRORS

After evaluation we looked the misclassified stories.

Most of stories which program classified as popular(created topic in Ekşi Sözlük), but not popular were actually popular stories, but Ekşi Sözlük users commented on persons topic who mentioned in stories, but did not create exclusive topic for it.

Also missclassified not popular stories were including a lot of popular terms or mentioned a lot of people/organization or location, but stories were cliche, so Ekşi Sözlük writers does not find it interesting.

References:

[1] <https://pdfs.semanticscholar.org/84d5/97ce7a56d1b53e0826261b357c3ce343c371.pdf>

[2] <https://github.com/hrzafer/resha-turkish-stemmer>

[3] <http://tools.nlp.itu.edu.tr/>

[4] <https://github.com/ironmanMA/Random-Forest>