Recognizing Complex Negation on Twitter

Junta Mizuno Canasai Kruengkrai Kiyonori Ohtake
Chikara Hashimoto Kentaro Torisawa Julien Kloetzer
National Institute of Information and Communications Technology
Kyoto 619-0289, Japan
{junta-m,canasai,kiyonori.ohtake,ch,torisawa,julien}@nict.go.jp
Kentaro Inui
Graduate School of Information Sceinces, Tohoku University
Miyagi 980-8579, Japan
inui@ecei.tohoku.ac.jp

Abstract

After the Great East Japan Earthquake in 2011, an abundance of false rumors were disseminated on Twitter that actually hindered rescue activities. This work presents a method for recognizing the *negation* of predicates on Twitter to find Japanese tweets that refute false rumors. We assume that the predicate “occur” is *negated* in the sentence “The guy who tweeted that a nuclear explosion occurred has watched too many SF movies.” The challenge is in the treatment of such *complex negation*. We have to recognize a wide range of complex negation expressions such as “it is theoretically impossible that...” and “The guy who... watched too many SF movies.” We tackle this problem using a combination of a supervised classifier and clusters of n-grams derived from large un-annotated corpora. The n-gram clusters give us a gain of about 22% in F-score for complex negations.

1 Introduction

After the Great East Japan Earthquake in 2011, hundreds of false rumors were disseminated on Twitter. At the same time, many experts and other knowledgeable people posted tweets to refute such false rumors as “There is no truth to the rumor that a nuclear explosion has occurred in Fukushima.” However, since many people did not notice such refutations, they retweeted the false rumors, inadvertently fueling the confusion and creating serious obstacles to rescue activities.

This paper presents a method that recognizes the *negation* of predicates on Twitter to identify the tweets that refute false rumors[1]. Our proposed method uses a supervised learning method to judge whether a given predicate in a tweet is negated.. An important point here is that we have to deal with *complex* forms of *negations* to achieve our final goal; the detection of false rumors. Note that although our target data are Japanese tweets, we provide examples in English for readability.

S1 It is theoretically impossible that a nuclear explosion *occurred* in Fukushima.

S2 The guy who tweeted that a nuclear explosion *occurred* has watched too many SF movies.

Both S1 and S2 refute that a nuclear explosion occurred. In other words, the predicate “occur” is negated, even though no negation words (e.g., “not”) are explicitly written in the sentences. Many sentences similar to the above examples were actually posted on Twitter to refute false rumors after the earthquake.

In this paper, we categorize negation into two types: simple and complex. Given a predicate that is annotated as negation by a human annotator, its categorization is done based on the following criteria.

Simple negation If at least one of the words in the same phrase of the negated predicate ends with “ない nai, わけない wakenai, ぬ nu (all of which mean not)” we define this form as simple negation. These three words are called *simple negation suffixes (SNW)* hereafter (Table 1) and roughly correspond to such simple forms of negations in English as “do not” and “has not.”

[1] Even though other linguistic expressions might also be useful to detect false rumors, we focus on negation.

29th Pacific Asia Conference on Language, Information and Computation pages 544 - 552
Shanghai, China, October 30 - November 1, 2015

Complex negation If a human annotator annotates a predicate as negated even without words that end with SNW in the same phrase of the negated predicate, we define this form as complex negation. For instance, the literal Japanese translations of S1 and S2 do not have any words that end with SNW.

We only focus on complex negation in this paper, since simple negation can be recognized by matching SNW against the words in the same phrase of the predicate with a high accuracy.

Thus, we have to recognize a wide range of expressions that indicate negation, including "it is theoretically impossible that..." and "The guy who... has watched too many SF movies." To tackle this problem, we use, as features for our supervised classifier, n-gram clusters derived from large unannotated corpora and generalize specific words or n-grams for them. Consider this sentence: "It is *untrue* that Kyoto has been heavily contaminated by radiation." If such a sentence exists in the training data for our classifier and the predicate "contaminated" is annotated as "negated," by the generalization of the word "*untrue*" to a cluster that includes "theoretically impossible," our method might successfully recognize that "occur" in S1 is negated. It also might even be possible to recognize the negation in S2 if several n-grams such as "guy," "too many," and "SF movies" are generalized to certain clusters and training samples can be found like "The people who claim that Tokyo was completely destroyed have watched too many Godzilla movies."

Through a series of experiments, we show that n-gram clusters give a 22% improvement in F-score for complex negations over the rule-based baseline. Our method successfully recognizes the complex forms of such negations as "An urban legend that..." and "Did dome idiot really say that...?".

To the best of our knowledge, this work is the first attempt to introduce n-gram clusters for recognizing complex forms of negation. Saurí (2008) developed a rule-based method to recognize the factuality of events, whose negation recognition can be regarded as a subtask. However, it seems quite difficult to write rules that cover the complex negation forms exemplified above. We expect that n-gram clusters play the role of the condition parts of the rules for complex negation forms.

Also note that we evaluate the performance of our method using the cross-validation on tweet data posted during one month immediately after the Great East Japan Earthquake. There is a possibility that this evaluation scheme may provide a high accuracy that cannot be achieved on real situation since the test data and training data were taken from the tweets concerning the same disaster. Nonetheless, it is difficult to provide test and training data concerning distinct large scale disasters. Therefore, we tried another setting in which the tweets posted during the two days immediately after the Great East Japan Earthquake were used as training data and the tweets posted after the first two days were used as test data. This evaluation scheme simulates the situation where new large scale disaster occurs and we have to prepare our system using the data available during first few days. We expect that the results give a lower bound of the performance of our method.

2 Related Work

Previous studies addressed negation recognition as part of modality/factuality analysis. Saurí and Pustejovsky (2009) annotated factuality for each event in TimeBank (Pustejovsky et al., 2003). Saurí and Pustejovsky (2007) defined three markers: polarity particles such as "not" or "no," modal particles such as "may" or "likely," and situation selecting predicates such as "prevent" or "suggest." They used these cue words to detect polarity (positive, negative, or unknown) and epistemic modality (certain, probable, possible, or unknown) and combined these two values to indicate an event's factuality.

de Marneffe et al. (2012) annotated the veridicality which roughly corresponds to the factuality for each event by ten annotators to the FactBank corpus (Saurí and Pustejovsky, 2009) and trained a classifier to predict the probabilistic distribution of event veridicality using a maximum entropy method (Berger et al., 1996). They compared the distributions predicted by the classifier and the distributions annotated by human annotators.

Soni et al. (2014) examined the factuality of quoted statements in tweets. They used the cue words defined in Saurí (2008) that introduced the quoted event negation or speculation and the tweet's author. They reported that conducting factuality analysis for quoted statements is quite difficult due to the error rate.

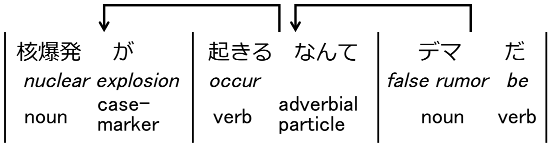

Figure 1: Japanese *bunsetsu* example

3 Approach

Our negation recognition algorithm takes an input sentence and classifies each predicate in it as "negated" or "non-negated." We train our classifier using support vector machines (SVMs) (Vapnik, 2000).

We use the notion of a *bunsetsu* which roughly corresponds to a *minimum* phrase in English and consists of a content words (basically nouns or verbs) and the functional words surrounding them. Figure 1 shows an example of a Japanese sentence, which means "That a nuclear explosion occurred is a false rumor." The vertical bars indicate the bunsetsu boundaries. A Japanese dependency tree is defined as a tree of the dependencies among the bunsetsus. In the above example, the first bunsetsu "核爆発_が kakubakuhatsu_ga (nuclear explosion)" depends on another bunsetsu "起きる_なんて okiru_nante (occur)," which in turn depends on "デマ_だ dema_da (be_false rumor)". The final bunsetsu is an exceptional case in which both a verb and a noun are included unlike the other bunsetsus that contain either a noun or a verb. Note that in this paper, bunsetsu boundaries and a dependency tree are given by J.DepP (Yoshinaga and Kitsuregawa, 2009).

3.1 Baseline Features

Basic Uni-, bi-, and tri-grams of words (surface, base form, and part of speech) in the bunsetsu include the target predicate and its head bunsetsu are used as basic features. The words in the two *bunsetsus* are distinguished in the feature set. If a *bunsetsu* includes a target predicate, n-grams are taken only from the strings following it. In the above example sentence, bi-gram "デマ_だ dema_da (be_false rumor)" and uni-gram "デマ dema (false rumor)" are included in this feature set when the target predicate is "起きる okiru (occur)."

Negation Words We manually created a short list of 33 words (CNW), such as "デマ dema (false rumor)" and "チェーンメール chainmail (chain letter)" shown in Table 2 that indicate complex forms of negations. These are often used to refute the information. Consider this tweet: "I have a chain letter that warns the occurrence of the explosion." The author of this tweet expresses his opinion that the explosion does not occur. For each word in the list, the features indicate the existence/non-existence of the word in the target sentence and its position ("before" or "after" the target predicate). They also include the distance of the negation words in the CNW from the target predicate. We encode the distance using 11-bit binary features (i.e., 1, 2, . . . , 10, or more). In the above example, the feature set encodes the information that the word "デマ dema (false rumor)" in the list is located after the target predicate "起きる okiru (occur)" along with the distance between "デマ dema (false rumor)" and "起きる okiru (occur)."

Sentiment We also consider the sentiment polarity of the words (positive or negative) in the *bunsetsus* from which the n-grams in the Basic feature set are taken. This feature set is useful because some words with negative polarities can express negation, like the word "ignorant" in "An *ignorant* person claimed a nuclear explosion actually occurred." We ignore words having neutral sentiment polarity. The words themselves with sentiment polarities are also encoded in this feature set. We use a list of words with manually annotated sentiment polarities (Oh et al., 2012).

Following Words In this feature set, we capture at most seven words (surface) that follow the target predicate, not including the target predicate. This feature is simple bag-of-words of uni-grams.

One crucial point is that we excluded from the feature set such *propositional contents* as "A nuclear explosion occurred," which is judged as "negated" in the S1 and S2 based on our criteria. This decision avoids the possibility that the classifier biases the negation of popular false rumors like the above nuclear explosion example. We assumed that propositional content which might be negated, is represented by a predicate and its argument and modifiers (and their descendants). Since Japanese is a head-final language, where a predicate appears at the right-hand side of its arguments and modifiers, we did not include the information concerning the left-hand side of a target predicate in the features and the target predicate itself[2]. In Figure 1, the predicate "起きる okiru (occur)" and its argument "核爆発 kakubakuhatsu (nuclear explosion)" are excluded.

[2] Except for the Sentiment Polarity features.

Synonyms of "not"	ない nai, ぬ nu, わけない wakenai

Table 1: List of simple negation suffixes (**SNW**)

Synonyms of "false rumor," "lie," "forgery" and "mistake"	デマ dema, でま dema, ガセ gase, ガセネタ gaseneta, がせ gase, ネタ neta, 風説 fusetsu, 流言 ryugen, 流言飛語 ryugenhigo, 流言蜚語 ryugenhigo, 誤報 goho, 誤情報 gojoho, 誤解 gokai, 嘘 uso, うそ uso, ウソ uso, 偽る itsuwaru, 偽り itsuwari, 捏造 netsuzo, ねつ造 netsuzo, 虚偽 kyogi, 間違う machigau, 間違い machigai, 出任せ demakase, でまかせ demakase, 誤る ayamaru, 誤り ayamari, 虚構 kyoko, 違う chigau, 違い chigai
Synonyms of "chain letter"	チェーンメール chainmail, チェンメ chenme, ちぇんめ chenme

Table 2: List of complex negation words (**CNW**)

3.2 N-Gram Cluster Features

A primary motivation behind the introduction of the n-gram cluster features is to *generalize* our CNW, which includes only 33 words. For instance, "デマ dema (false rumor)" has many synonymous expressions such as "偽_情報 nise_joho (false information)", and "不確かな_情報 futashikana_joho, 不正確な_情報 fuseikakuna_joho, 不確定な_情報 fukakuteina_joho, 真偽_不明な_情報 shingi_humeina_joho (all of which mean uncertain information)", which are not covered by CNW.

We used an implementation of a neural network-based algorithm (i.e., word2vec[3] (Mikolov et al., 2013)) to construct the synonym clusters. To get larger units of words (i.e. n-grams), we ran the word2phrase tool on these corpora twice and generated n-gram clusters by performing the k-means clustering algorithm on the top of the word (and phrase) vectors. This feature set encodes the semantic cluster IDs of the words and the n-grams ($n \leq 4$) found in the seven words that follow a target predicate. Note that we modified the k-means clustering of the word2vec tool so that the word vectors are normalized to the length of the vector.

Three distinct corpora were given to the word2vec tool:

1. All of the articles from Japanese Wikipedia (revision of 18 Jan. 2015),
2. Web pages crawled in 2007, i.e., about four years before the earthquake,
3. Twitter data posted from Feb. 14 to 28, 2015, i.e., about four years after the earthquake.

We randomly sampled sentences for corpora 2 (4.5 GB) and 3 (4.3 GB) to match Wikipedia's size (4.2 GB). We tokenized each document with a morphological analyzer MeCab (Kudo et al., 2004) and the Juman PoS tag set (Kurohashi et al., 1994) and applied the word2vec tool and k-means clustering. We needed to choose several parameters, including the numbers of vector dimensions and clusters, whose values were based on 5-fold cross-validation on our annotated training data, as described in Section 4.1. We tried eight dimensions (50, 100, 150, 200, 250, 300, 350, and 500) of vectors and six numbers of clusters (100, 500, 1,000, 2,000 5,000, and 10,000) for each corpus. For the optimal parameters, which worked best in our preliminary experiments, we finally chose 250 as the dimension and 10,000 as the number of clusters for Wikipedia, 350 dimensions and 10,000 clusters for the Web, and 300 dimensions and 10,000 number of clusters for Twitter. The tuning for the word2vec parameters was done by using a classifier with the word2vec features and the Basic, Negation Words, and Sentiment features for our classifiers.

In our experiments, many synonymous expressions such as "偽_情報 nise_joho (false information)" and "不確かな_情報 futashikana_joho, 不正確な_情報 fuseikakuna_joho, 不確定な_情報 fukakuteina_joho, 真偽_不明な_情報 shingi_fumeina_joho (all of which mean uncertain information)" were assigned vectors close to that of "デマ dema (false rumor)" in terms of cosine similarity. We assume that the clusters obtained by k-means might capture such synonyms, i.e., the cluster including "デマ dema (false rumor)" also includes its synonyms. CNW also includes only single words, while performing k-means clusters on word/phrase vectors can assign cluster IDs to n-grams. Actually, in the above examples the expression "不確かな_情報 futashikana_joho (uncertain information)" consists of two words. Furthermore, we assume that the combination of a supervised classifier and n-gram clusters can capture, to a certain degree, extremely complex negation forms, such as the negations of "occur" in the sentence, "The guy who tweeted that a nuclear explosion occurred has watched too many SF movies" by such clusters including n-grams as "guy," "too many," and "SF movies."

[3] https://code.google.com/p/word2vec/

Usage	Source	#predicates	#nps	#cns
training	tweets	96,824	11,842	1,541
	artificial	4,048	1,638	849
	total	100,872	13,480	2,390
test	tweets	14,253	2,250	393

Table 3: The training and test sets. **#nps** indicates number of negation instances and **#cns** indicates number of complex negated instances.

4 Experiments

4.1 Experimental Settings

We first asked human annotators to judge whether 115,125 predicate instances sampled from tweets were negated. Table 3 shows the number of instances in the training and test sets. Instances whose source is tweets in the table were extracted from tweets posted during within one month after the Great East Japan Earthquake (from March 11 2011 to April 11 2011) and instances whose source is artificial in the table were manually composed of tweet-like texts that included typical examples of complex forms of negations to expand the number of complex negations. Note that we also used the training set as the development set for parameter tuning by 5-fold cross-validation. All of the test set instances were extracted from tweets and there was no overlap between the training and test sets.

In both sets, each predicate was annotated by a single annotator by the following steps:

1. We annotated predicates based on Iida et al. (2007). All of the verbs and adjectives were annotated as predicates, and some nouns were annotated as nominal predicates.
2. We annotated negation by the negation cues surrounding the predicate. Both such functional expressions as "ない nai (not)" and such content words as "嘘_だ uso_da (it is doubtful that)" are used as negation cues.
3. The predicates (interpreted by the annotator as negated by the cues) are annotated as negated predicates and used as positive instances for SVM, and the others are used as negative instances.

Note that recognizing negated instances as either simple or complex is done automatically based on the definitions described in Section 1.

In all the experiments, we used LIBSVM (Chang and Lin, 2011) with a degree 2 polynomial kernel, where gamma = 1 and cost = 0.001, which worked best in our preliminary experiments. We set the remaining parameters to the tool's default values.

4.2 Baseline Methods

We conducted experiments using three baselines and created two baseline systems built on rule- and machine learning-based methods. We also adapted a method (de Marneffe et al., 2012) that recognizes veridicality and negation.

Rule is a simple rule-based method that regards a predicate as "negated" only when any of the negation words in Tables 1 and 2 are found in a window of seven words on each side of the target predicate as well as the target predicate itself.

ML uses the SVM classifier with the four features described in Section 3: basic, negation words, sentiment, and following words.

Marneffe12 predicts the veridicality of the propositions written in a sentence as well as the negation recognition. We replicated their features, except the "world knowledge" feature that captures the subject of the target predicate. In the following, we describe how to apply these features to Japanese.

Predicate classes Some words are often used to introduce the factuality of events. For instance, given "He confirmed that she will come," "confirmed" indicates that the factuality of "come" is certainty. We translated 779 words with 38 classes (Saurí, 2008) into 4,110 words in Japanese. We used the class name and the original form of the word as binary features to detect whether the target predicate is led by one of these words.

General features We used the original forms of the predicate and the sentence's root.

Modality features We identified such modal expressions as "かも kamo (might)" in two *bunsetsus* in a dependency, where the head *bunsetsu* contains the target predicate. The other modal expressions found elsewhere in the sentence are marked as different features.

Negation We used both SNW and CNW to find negation words.

Conditional We examined whether the predicates are in an *if*-clause and checked whether they end with "たら tara (if)" or "れば reba (if)" and words indicating *if*-clauses such as "もし moshi (if)" and "仮に karini (if)."

Quotation We also checked whether the sentence opened and ended with quotation marks.

Originally, de Marneffe et al. (2012) used the maximum entropy classifier (Berger et al., 1996).

Method	Precision (%)		Recall (%)		F-score (%)
Rule	8.01	(1071 / 13377)	44.81	(1071 / 2390)	13.59
Marneffe12	14.62	(371 / 2538)	15.52	(371 / 2390)	15.06
ML	77.43	(621 / 802)	25.98	(621 / 2390)	38.91
ML + web-d350	76.84	(617 / 803)	25.82	(617 / 2390)	38.65
ML + wikipedia-d250	76.89	(632 / 822)	26.44	(632 / 2390)	39.35
ML + twitter-d300	77.28	(643 / 832)	26.90	(643 / 2390)	39.91
ML + all	71.40	(839 / 1175)	35.10	(839 / 2390)	47.07
Proposed	69.81	(867 / 1242)	36.28	(867 / 2390)	**47.74**

Table 4: Results on the training set by 5-fold cross-validation

Method	Precision (%)		Recall (%)		F-score (%)
web d300 n2000	78.09	(613 / 785)	25.65	(613 / 2390)	38.61
wikipedia d500 n2000	77.42	(617 / 797)	25.82	(617 / 2390)	38.72
twitter d200 n2000	77.56	(622 / 802)	26.03	(622 / 2390)	38.97
noun-cls n2000	77.12	(691 / 896)	28.91	(691 / 2390)	42.06

Table 5: Comparison to other clustering methods on the training set by 5-fold cross-validation

We used SVM as a classifier with the same parameters as our proposed method described in Section 3.

4.3 Proposed Methods

In our proposed method, we used all of the features described in Section 3 along with all of the cluster IDs obtained using the Wikipedia, Tweets, and Web sets. Table 4 shows the results of our proposed methods, some baselines, and our method without certain types of features on the training set by 5-fold cross-validation. Although both simple and complex negations are used as positive instances for SVM, we evaluated only complex negations.

The comparison between "Rule" and "ML" suggests that a predicate is not always negated even if it is surrounded by negation words. There are two reasons for poor performance of "Rule." The first is false matching of simple negation words for idiomatic expressions. For instance, "思いがけ_ない omoigake_nai (unexpected)" contains the negation word "ない nai (not)" at the end of the word but it does not express negation in Japanese. The second is a double negative. For instance, "間違い_ない machigai_nai (it is not incorrect)" contains two negation words "間違う machigau (incorrect)" and "ない nai (not)" but it does not express negation and roughly corresponds to "it is correct." The comparison between "Marneffe12" and "ML" suggests that it is infeasible to cover various negation words and their n-grams using word lists organized manually.

The "ML+web-d350," "ML+wikipedia-d250," and "ML+twitter-d300" columns indicate the performance of using the n-gram cluster features with three types of corpora. The n-gram cluster feature generated from Twitter outperforms the other two corpora. The "ML+all" column indicates using three corpora at once. These features are distinguished by their sources. It outperforms the other settings of using each corpora. The comparisons between the "ML" and "ML+all" and "Marneffe12" and "ML+all" suggest that n-gram clusters successfully generalize complex negations forms by their cluster IDs.

We compare our n-gram clustering method with "noun-cls," another clustering method that was proposed by Kazama and Torisawa (2008). We applied their clustering algorithm to nouns extracted from roughly six hundred million Web documents. The Web documents that we used for our clustering are a subset of these documents. The variation of words in the noun-cluster is wider than in other clusters. We set the clustering number to 2,000 in our n-gram clustering method and "noun-cls." Table 5 compares the clustering method and the corpora that we used. We tried eight dimensions (50, 100, 150, 200, 250, 300, 350, and 500), and the number of dimensions in Table 5 achieved the best performance for each corpus. The "noun-cls" column outperformed the other clusters. This suggests that generalization by noun-cluster is more effective than n-gram clusters for complex negation recognition because the noun-cluster has more various words than the others. In future work, we will generate n-gram clusters from large-scale documents.

4.4 Results on the Test Set

We trained a classifier using the whole training set as training data and applied it to our test set. Table 6 shows the performance of the following seven methods. Here, "Proposed" indicates the performance of our proposed method, and "Rule" is the perfor-

Method	**Precision (%)**	**Recall (%)**	**F-score (%)**
Proposed	64.04 (114 / 178)	29.01 (114 / 393)	**39.93**
Rule	10.43 (220 / 2109)	55.98 (220 / 393)	17.59
ablation test			
$-n$-gram-cls	71.57 (73 / 102)	18.58 (73 / 393)	29.49
$-$noun-cls	62.96 (102 / 162)	25.95 (102 / 393)	36.76
$-$basic	73.04 (84 / 115)	21.37 (84 / 393)	33.07
$-$following	66.23 (100 / 151)	25.45 (100 / 393)	36.76
$-$negation	63.25 (105 / 166)	26.72 (105 / 393)	37.57
$-$sentiment	61.59 (101 / 164)	25.70 (101 / 393)	36.27
$-n$-gram-cls+uni-gram-cls	63.75 (102 / 160)	25.95 (102 / 393)	36.89

Table 6: Results on the test set

納豆が放射能に**効く**という都市伝説があったりも There's also this urban legend saying that natto (Japanese food made from fermented soybeans) **is effective** against radiation.
避難所で物資横流しが**起きてる**ってツイートしてる奴、北斗の拳の見すぎ〜 The guy who tweeted that there **is some supplies black market** at the shelters, (I'm sure) he watched too much of "Fist of the North Star" (Japanese dystopic SF animation)
国会議事堂が**破壊された**なんてほざいてるバカいるの Is there really an idiot who said that the National Diet (building) **was destroyed**?

Table 7: Examples of output

mance of a rule-based baseline method, which regards a predicate as "negated" only when any words of SNW and CNW in Tables 1 and 2 are found in a 14-word window centered on the predicate. The proposed method achieved more than 20% improvement in F-score over the rule-based method for complex negations.

Our ablation test on the test set is shown in "ablation test." For each result, one of the features is ablated. The "$-n$gram-cls+uni-gram-cls" column indicates that we only generalized a uni-gram (i.e., single word). This setting confirms the effect of n-gram generalization. In the ablation test, the results indicate that every feature was effective. In other words, lower performance was observed after removing each feature. The comparison between the proposed method and the method without cluster IDs "-n-gram-cls" suggests that cluster IDs are useful for recognizing complex types of negations. The comparison between the proposed method and the method without 2,3,4-gram generalization "$-n$-gram-cls+uni-gram-cls" suggests that n-gram generalization is effective for complex negation recognition. For instance, an n-gram cluster of Wikipedia has such uni-grams as "不正確である fuseikakudearu (incorrect)" and "不完全である fukanzendearu (incomplete)" as synonyms of bi-gram "正しく_ない tadashiku_nai (not correct)."

We also show in Table 7 three negated predicates extracted from real tweets that were not properly recognized by either the rule-based method or the machine learning method without the n-gram clusters, but they were correctly classified by our method.

5 Simulation of Disaster Situation

We constructed our data set from tweets in the month following the Great East Japan Earthquake. Since that the purpose of negation recognition is to detect false rumors during a disaster, taking one month to annotate the data and to train classifiers is too long. Therefore, we simulated the situation as in Figure 2.

Before 3/10	We do not have any annotated corpus.
14:46 at 3/11	The earthquake occurred. Start annotation.
14:00 at 3/13	Stop annotation. Now we have annotated the data extracted from tweets posted two days just after the earthquake.
After 14:00 at 3/13	Start negation recognition using the trained classifier with the annotated data.

Figure 2: Simulation settings

We re-organized the composition of the training and the test sets as follows:

- Training set 2 contains 626 complex negation instances extracted from tweets posted from 14:00 3/11 to 14:00 3/13,
- Test set 2 contains 1,269 complex negation instances extracted from tweets posted after 14:00 3/13.

Note that some tweets have been posted before 14:00 3/11 and they are not used in this experiments.

In this case, we should make the classifier more specific to the disaster that actually occurred; the previous experiments considered a general situation. We experimented with two approaches.

Method	Precision (%)	Recall (%)	F-score (%)
Proposed	48.96 (212 / 433)	16.71 (212 / 1269)	24.91
+twitter2days	46.41 (220 / 474)	17.34 (220 / 1269)	25.24
+content	52.49 (253 / 482)	19.94 (253 / 1269)	28.90
+twitter2days+content	51.64 (268 / 519)	21.12 (268 / 1269)	**29.98**

Table 8: Results of disaster situation simulation

The first approach obtained another set of n-gram clusters using all the tweets posted in the two days (4.8GB) denoted by twitter2days. We set the number of vector dimensions to 300 and the number of clusters to 10,000. This set of clusters has more specific synonymous words than the other clusters obtained for tweets not in the disaster. For instance, “雨 ame (rain)” has many expressions that are synonyms only in this disaster, such as “汚染_さ_れた_雨 osen_sa_reta_ame, 黒い_雨 kuroi_ame, 有害な_雨 yugaina_ame, 有毒な雨 yudokuna_ame (all of which mean toxic rain),” while in other clusters “雨 ame (rain)” has general synonyms, such as “小雨 kosame (light rain),” “冷たい_雨 tsumetai_ame (cold rain),” and “大雨 ohame (heavy rain).” These specific synonyms were obtained because the following false rumor was disseminated and repeated for a long period: “There may be toxic rain due to an explosion at Cosmo Oil.”

The second approach uses target predicates and their arguments for feature generation. Some false rumors were disseminated and repeated for a long periods, such as “Drinking iodine protects against radiation.” There were also many tweets to negate such false rumors. Therefore, we used the content of the false rumors for training to recognize other negated predicates whose content is the same as the trained ones. We modified the “following words” and “n-gram clusters” features to use not only the seven words that follow the target predicate but also the target predicate itself.

We had to choose two parameters: the number of vector dimensions and the number of clusters for four corpora: web, wikipedia, twitter, and twitter2days. We chose the same numbers of the previous experiment for the three former corpora and chose 300 as the dimension and 10,000 as the number of clusters for the last one that is identical to twitter in the general situation.

Table 8 shows the results. The “+twitter2days d300 n10000” column indicates that we used the extra n-gram cluster and outperformed “Proposed,” which had the best setting in the previous experiments. Even if we have a limited amount of annotated data in the disaster, large un-annotated corpora can improve the performance of complex negation recognition. Note that the performance of complex negation recognition is lower than the previous experiments since we used a smaller annotated corpus in this simulation.

The “+content” column indicates that we modified the features to capture the content of the predicate, and the “+twitter2days+content” column indicates that we used the extra n-gram cluster with the content features. The comparisons between “Proposed” and “+content” and “+twitter2days” and “+twitter2days+content” suggest that when many tweets are disseminated and repeated for a long periods about particular topics, we must use content words.

6 Conclusion

We presented a method for recognizing negations on Twitter and showed that n-gram clusters derived from large un-annotated corpora obtained by word2vec are effective for capturing complex types of negations, like the negations of “occur” in the sentence “The guy who tweeted that a nuclear explosion occurred has watched too many SF movies.” We also simulated the situation of the Great East Japan Earthquake in 2011. We used annotated data posted within two days after the earthquake for training, and we also recognized negation for tweets posted on other days. We found that using un-annotated data for the “n-gram clusters” feature and the capturing contents are effective for negation recognition. We are going to implement a false rumor detection system by integrating our proposed method with the rule-based method. We expect our system to be useful in future disaster situations.

Acknowledgments

This work was partially supported by the Council for Science, Technology and Innovation (CSTI) through the Cross-ministerial Strategic Innovation Promotion Program (SIP), titled “Enhancement of societal resiliency against natural disasters” (Funding agency: JST).

References

Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Marie-Catherine de Marneffe, Christopher D. Manning, and Christopher Potts. 2012. Did it happen? The pragmatic complexity of veridicality assessment. *Computational Linguistics*, 38(2):301–333.

Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. 2007. Annotating a Japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the Linguistic Annotation Workshop*, pages 132–139.

Jun'ichi Kazama and Kentaro Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *Proceedings of ACL-08: HLT*, pages 407–415, Columbus, Ohio, June. Association for Computational Linguistics.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *EMNLP*, volume 4, pages 230–237.

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of The International Workshop on Sharable Natural Language*, pages 22–28.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Jong-Hoon Oh, Kentaro Torisawa, Chikara Hashimoto, Takuya Kawada, Stijn De Saeger, Jun'ichi Kazama, and Yiou Wang. 2012. Why question answering using sentiment analysis and word classes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 368–378. Association for Computational Linguistics.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40.

Roser Saurí and James Pustejovsky. 2007. Determining modality and factuality for text entailment. In *First IEEE International Conference on Semantic Computing*, pages 509–516. IEEE.

Roser Saurí and James Pustejovsky. 2009. FactBank: A corpus annotated with event factuality. *Language resources and evaluation*, 43(3):227–268.

Roser Saurí. 2008. *A factuality profiler for eventualities in text*. Ph.D. thesis, Brandeis University.

Sandeep Soni, Tanushree Mitra, Eric Gilbert, and Jacob Eisenstein. 2014. Modeling factuality judgments in social media text. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 415–420, Baltimore, Maryland, June. Association for Computational Linguistics.

Vladimir Vapnik. 2000. *The nature of statistical learning theory*. Springer Science & Business Media.

Naoki Yoshinaga and Masaru Kitsuregawa. 2009. Polynomial to linear: Efficient classification with conjunctive features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1542–1551.