Using Sequence Kernels to identify Opinion Entities in Urdu

Smruthi Mukund[†] **and Debanjan Ghosh**[*]
[†]SUNY at Buffalo, NY
smukund@buffalo.edu
[*]Thomson Reuters Corporate R&D
debanjan.ghosh@thomsonreuters.com

Rohini K Srihari
SUNY at Buffalo, NY
rohini@cedar.buffalo.edu

Abstract

Automatic extraction of opinion holders and targets (together referred to as opinion entities) is an important subtask of sentiment analysis. In this work, we attempt to accurately extract opinion entities from Urdu newswire. Due to the lack of resources required for training role labelers and dependency parsers (as in English) for Urdu, a more robust approach based on (i) generating candidate word sequences corresponding to opinion entities, and (ii) subsequently disambiguating these sequences as opinion holders or targets is presented. Detecting the boundaries of such candidate sequences in Urdu is very different than in English since in Urdu, grammatical categories such as tense, gender and case are captured in word inflections. In this work, we exploit the morphological inflections associated with nouns and verbs to correctly identify sequence boundaries. Different levels of information that capture context are encoded to train standard linear and sequence kernels. To this end the best performance obtained for opinion entity detection for Urdu sentiment analysis is 58.06% F-Score using sequence kernels and 61.55% F-Score using a combination of sequence and linear kernels.

1 Introduction

Performing sentiment analysis on newswire data facilitates the development of systems capable of answering perspective questions like "*How did people react to the latest presidential speech?*" and "*Does General Musharraf support the Indo-Pak peace treaty?*". The components involved in developing such systems require accurate identification of opinion expressions and opinion entities. Several of the approaches proposed in the literature to automatically extract the opinion entities rely on the use of thematic role labels and dependency parsers to provide new lexical features for opinion words (Bethard *et al.*, 2004). Semantic roles (SRL) also help to mark the semantic constituents (*agent*, *theme*, *proposition*) of a sentence. Such features are extremely valuable for a task like opinion entity detection.

English is a privileged language when it comes to the availability of resources needed to contribute features for opinion entity detection. There are other widely spoken, resource poor languages, which are still in the infantile stage of automatic natural language processing (NLP). Urdu is one such language. The main objective of our research is to provide a solution for opinion entity detection in the Urdu language. Despite Urdu lacking NLP resources required to contribute features similar to what works for the English language, the performance of our approach is comparable with English for this task (compared with the work of Weigand and Klalow, 2010 ~ 62.61% F1). The morphological richness of the Urdu language enables us to extract features based on noun and verb inflections that effectively contribute to the opinion entity extraction task. Most importantly, these features can be generalized to other Indic languages (Hindi, Bengali etc.) owing to the grammatical similarity between the languages.

Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pages 58–67,
Portland, Oregon, USA, 23–24 June 2011. ©2011 Association for Computational Linguistics

English has seen extensive use of sequence kernels (string and tree kernels) for tasks such as relation extraction (Culotta and Sorensen, 2004) and semantic role labeling (Moschitti *et al.*, 2008). But, the application of these kernels to a task like opinion entity detection is scarcely explored (Weigand and Klalow, 2010). Moreover, existing works in English perform only opinion holder identification using these kernels. What makes our approach unique is that we use the power of sequence kernels to simultaneously identify opinion holders and targets in the Urdu language.

Sequence kernels allow efficient use of the learning algorithm exploiting massive number of features without the traditional explicit feature representation (such as, Bag of Words). Often, in case of sequence kernels, the challenge lies in choosing meaningful subsequences as training samples instead of utilizing the whole sequence. In Urdu newswire data, generating candidate sequences usable for training is complicated. Not only are the opinion entities diverse in that they can be contained within noun phrases or clauses, the clues that help to identify these components can be contained within any word group - speech events, opinion words, predicates and connectors.

1	***Pakistan ke swaat sarhad ke janoobi shahar Banno ka havayi adda*** *zarayye ablaagk tavvju ka markaz ban gaya hai.* *[**Pakistan's provincial border's south city's airbase** has become the center of attraction for all reporters.]* Here, the opinion target spans across four noun chunks, "**Pakistan's | provincial border's | south city's | airbase**". The case markers (connectors) ***"<u>ke</u>"***and***"<u>ka</u>"*** indicate the span.
2	***Habib miyan ka*** *ghussa bad gaya aur wo* ***apne aurat*** *ko* ***maara****.* ***[Habib miya's*** *anger increased and he* ***hit*** *his own* ***wife****.]* Here, the gender (<u>*Masculine*</u>) inflection of the verb ***"maara"*** *(hit)* indicates that the agent performing this action is **"Habib miya"** (<u>*Masculine*</u>)
3	*Ansari ne* ***kaha*** *"**mere rayee mein** Aamir Sohail eek badimaak aur Ziddi insaan hai".* *[Ansari said, "**according to me** Aamir Sohail is one crazy and stubborn man"]* Here, cues similar to English such as "***mere rayee mein****" (according to)* indicate the opinion holder. Another interesting behavior here is the presence of nested opinion holders. ***"kaha"*** *(said)* indicates that this statement was made by Ansari only.
4	*Sutlan bahut khush tha,* ***naseer key kaam se****.* *[Sultan was very happy with* ***Naseer's work****]* Here, the target of the expression "***khush"*** is after the verb ***"khush tha"(****was happy) – SVO structure*

Table 1: Examples to outline the complexity of the task

Another contributing factor is the free word order of the Urdu language. Although the accepted form is SOV, there are several instances where the object comes after the verb or the object is before the subject. In Urdu newswire data, the average number of words in a sentence is 42 (Table 3). This generates a large number of candidate sequences that are not opinion entities, on account of which the data used for training is highly unbalanced. The lack of tools such as dependency parsers makes boundary detection for Urdu different from English, which in turn makes opinion entity extraction a much harder task. Examples shown in table 1 illustrate the complexity of the task.

One safe assumption that can be made for opinion entities is that they are always contained in a phrase (or clause) that contains a noun (common noun, proper noun or pronoun), which is either the subject or the object of the predicate. Based on this, we generate candidate sequences by considering contextual information around noun phrases. In example 1 of Table 1, the subsequence that is generated will consider all four noun phrases **"Pakistan's | provincial border's | south city's | airbase"** as a single group for opinion entity.

We demonstrate that investigating postpositions to capture semantic relations between nouns and predicates is crucial in opinion entity identification. Our approach shows encouraging performance.

2 Related Work

Choi *et al.*, (2005) consider opinion entity identification as an information extraction task and the opinion holders are identified using a conditional random field (Lafferty *et al.,* 2001) based sequence-labeling approach. Patterns are extracted using AutoSlog (Riloff *et al.,* 2003). Bloom *et al.,* (2006) use hand built lexicons for opinion entity identification. Their method is dependent on a combination of heuristic shallow parsing and dependency parsing information. Kim and Hovy

(2006) map the semantic frames of FrameNet (Baker *et al.,* 1998) into opinion holder and target for adjectives and verbs to identify these components. Stoyanov and Cardie (2008) treat the task of identifying opinion holders and targets as a co-reference resolution problem. Kim *et al.,* (2008) used a set of communication words, appraisal words from Senti-WordNet (Esuli and Sebastiani, 2006) and NLP tools such as NE taggers and syntactic parsers to identify opinion holders accurately. Kim and Hovy (2006) use structural features of the language to identify opinion entities. Their technique is based on syntactic path and dependency features along with heuristic features such as topic words and named entities. Weigand and Klalow (2010) use convolution kernels that use predicate argument structure and parse trees.

For Urdu specifically, work in the area of classifying subjective and objective sentences is attempted by Mukund and Srihari, (2010) using a vector space model. NLP tools that include POS taggers, shallow parser, NE tagger and morphological analyzer for Urdu is provided by Mukund *et al.*, (2010). This is the only extensive work done for automating Urdu NLP, although other efforts to generate semantic role labels and dependency parsers are underway.

3 Linguistic Analysis for Opinion Entities

In this section we introduce the different cues used to capture the contextual information for creating candidate sequences in Urdu by exploiting the morphological richness of the language.

Case	Clitic Form	Examples
Ergative	(ne)	*Ali* ***ne*** *ghussa dikhaya ~ Ali showed anger*
Accusative	(ko)	*Ali* ***ko*** *mainey maara ~ I hit Ali*
Dative	(ko,ke)	Similar to accusative
Instrumental	(se)	*Yeh kaam Ali* ***se*** *hua ~ This work was done by Ali*
Genitive	(ka, ke, ki)	*Ali* ***ka*** *ghussa, baap re baap! ~ Ali's anger, oh my God!*
Locative	(mein, par, tak, tale, talak)	*Ali* ***mein*** *ghussa zyaada hai ~ there is a lot of anger in Ali*

Table 2: Case Inflections on Nouns

Urdu is a head final language with post-positional case markers. Some post-positions are associated with grammatical functions and some with specific roles associated with the meaning of verbs (Davison, 1999). Case markers play a very important role in determining the case inflections of nouns. The case inflections that are useful in the context of opinion entity detection are *"ergative"*, *"dative"*, *"genitive"*, *"instrumental"* and *"locative"*. Table 2 outlines the constructs.

Consider example 1 below. (a) is a case where "*Ali*" is nominative. However, in (b) *"Ali"* is dative. The case marker *"ko"* helps to identify subjects of certain experiential and psychological predicates: sensations, psychological or mental states and obligation or compulsion. Such predicates clearly require the subject to be sentient, and further, indicate that they are affected in some manner, correlating with the semantic properties ascribed to the dative's primary use (Grimm, 2007).

Example (1):

(a) Ali khush ***hua*** *(Ali became happy)*
(b) Ali ***ko*** *khushi* ***hui*** *(Ali became happy)*

Example (2):

(a) ***Sadaf*** *kaam karne ki koshish* ***karti hai*** *(Sadaf tries to do work)*

Semantic information in Urdu is encoded in a way that is very different from English. Aspect, tense and gender depend on the noun that a verb governs. Example 2 shows the dependency that verbs have on nouns without addressing the linguistic details associated with complex predicates.

In example 2, the verb *"karti"(do)* is *feminine* and the noun it governs ~*Sadaf* is also *feminine*. The doer for the predicate *"karti hai"(does)* is *"Sadaf"* and there exists a gender match. This shows that we can obtain strong features if we are able to accurately (i) identify the predicates, (ii) find the governing noun, and (iii) determine the gender.

In this work, for the purpose of generating candidate sequences, we encompass the post-position responsible for case inflection in nouns, into the noun phrase and group the entire chunk as one single candidate. In example 1, the dative inflection on *'Ali'* is due to the case marker *'ko'*. Here, *'Ali ko'* will always be considered together in all candidate sequences that this sentence generates. This behavior can also be observed in example 1 of table 1.

We use Semantex^{TM} (Srihari *et al.*, 2008) - an end to end NLP framework for Urdu that provides POS, NE, shallow parser and morphological analyzer, to mark tense, mood, aspect, gender and number inflections of verbs and case inflections of nouns. For ease of parsing, we enclose dative and accusative inflected nouns and the respective case markers in a tag called *POSSESS*. We also enclose locative, genitive and ergative inflections and case markers in a tag called *DOER*.

4 Methodology

Sequence boundaries are first constructed based on the POSSESS, DOER and NP (noun chunk) tags prioritized by the position of the tag while parsing. We refer to these chunks as *"candidates"* as they are the possible opinion entity candidates. We generate candidate sequences by combining these candidates with opinion expressions (Mukund and Srihari, 2010) and the predicates that contain or follow the expression words (~*khushi* in (b) of example 1 above).

We evaluate our approach in two steps:

(i) Boundary Detection - detecting opinion entities that contain both holders and targets

(ii) Entity Disambiguation - disambiguating opinion holders from opinion targets

In the following sections, we briefly describe our research methodology including sequence creation, choice of kernels and the challenges thus encountered.

4.1 Data Set

The data used for the experiments are newswire articles from BBC Urdu[1] that are manually annotated to reflect opinion holders, targets, and expressions (emotion bearing words).

Number of subjective sentences	824
Average word length of each sentence	42
Number of opinion holders	974
Number of opinion targets	833
Number of opinion expressions	894

Table 3: Corpus Statistics

Table 3 summarizes the corpus statistics. The inter annotator agreement established between two annotators over 30 documents was found to be 0.85 using Cohen's Kappa score (averaged over all tags). The agreement is acceptable as tagging emotions is a difficult and a personalized task.

4.2 Support Vector Machines (SVM) and Kernel Methods

SVMs belong to a class of supervised machine learning techniques that merge the nuances of statistical learning theory, kernel mapping and optimization techniques to discover separating hyperplanes. Given a set of positive and negative data points, based on structural risk minimization, SVMs attempt to find not only a separating hyperplane that separates two categories (Vapnik and Kotz, 2006) but also maximize the boundary between them (maximal margin separation technique). In this work, we propose to use a variation of sequence kernels for opinion entity detection.

4.3 Sequence Kernels

The lack of parsers that capture dependencies in Urdu sentences inhibit the use of 'tree kernels' (Weigand and Klalow, 2010). In this work, we exploit the power of a set of sequence kernels known as 'gap sequence string kernels' (Lodhi *et al.*, 2002). These kernels provide numerical comparison of phrases as entire sequences rather than a probability at the chunk level. Gap sequence kernels measure the similarity between two sequences (in this case a sequence of Urdu words) by counting the number of common subsequences. Gaps between words are penalized with suitable use of decay factor $(\lambda; 0 < \lambda < 1)$ to compensate for matches between lengthy word sequences.

Formally, let Σ_i be the feature space over words. Consequently, we declare other disjoint feature spaces $\Sigma_j, \Sigma_k ... \Sigma_l$ (stem words, POS, chunks, gender inflections, etc.) and $\Sigma_x = \Sigma_i \times \Sigma_j \times \Sigma_k ... \Sigma_l$. For any two-feature vectors $s, t \in \Sigma_x$ let $f(s,t)$ compute the number of common features between *s* and *t*. Table 5 lists the features used to compute $f(s,t)$.

Given two sequences, *s* and *t* and the kernel function $K_{es}(s,t,\lambda)$ that calculates the number of

[1] www.bbc.co.uk/urdu/

weighted sparse subsequences of length *n* (say, *n*=2: bigram) common to both *s* and *t*, then $K_{es}(s,t,\lambda)$ is as shown in eq 1 (Bunescu and Mooney, 2005).

$$K_{es}(s,t,\lambda) = \sum_{\mathbf{i}:|\mathbf{i}|=n} \sum_{\mathbf{j}:|\mathbf{j}|=n} \prod_{k=1}^{n} s(s_{i_k}, t_{j_k}) \lambda^{l(\mathbf{i})+l(\mathbf{j})}$$

(i,j,k are dimensions) ------ Eq 1.

Generating correct sequences is a prior requirement for sequence kernels. For example, in the task of relation extraction, features included in the shortest path between the mentions of the two sequences (which hold the relation) play a decisive role (Bunescu and Mooney, 2005). Similarly, in the task of role labeling (SRL - Moschitti *et al.*, 2008), syntactic sub-trees containing the arguments are crucial in finding the correct associations. Our approach to create candidate sequences for opinion entity detection in Urdu is explained in the next section.

4.4 Candidate Sequence Generation

Each subjective sentence in Urdu contains several noun phrases with one or more opinion expressions. The words that express opinions (expression words) can be contained within a verb predicate (if the predicate is complex) or precede the verb predicate. These subjective sentences are first pre-processed to mark the morphological inflections as mentioned in §3.

1	A sentence is parsed to extract all likely candidate chunks – POSSESS, DOER, NP in that order.
2	<expression, predicate> tuples are first selected based on nearest neighbor rule : 1. Predicates that are paired with the expression words either contain the expressions or follow the expressions. 2. Stand alone predicates are simply ignored as they do not contribute to the holder identification task (they contribute to either the sentence topic or the reason for the emotion).
3	For each candidate, <candidate, expression, predicate> tuples are generated without changing the word order. (Fig. 1 – example candidates maintain the same word order)

Table 4: Candidate Sequence Generation

We define training candidate sequences as the shortest substring *t* which is a tuple that contains the candidate noun phrase (POSSESS, DOER or NP), an emotion expression and the closest predicate. Table 4 outlines the steps taken to create the candidate sequences and figure 1 illustrates the different tuples for a sample sentence.

Experiments conducted by Weigand and Klakow (2010) consider <candidate, predicate> and <candidate, expression> tuples. However, in Urdu the sense of expression and predicate are so tightly coupled (in many examples they subsume each other and hence inseparable), that specifically trying to gauge the influence of predicate and expression separately on candidates is impossible.

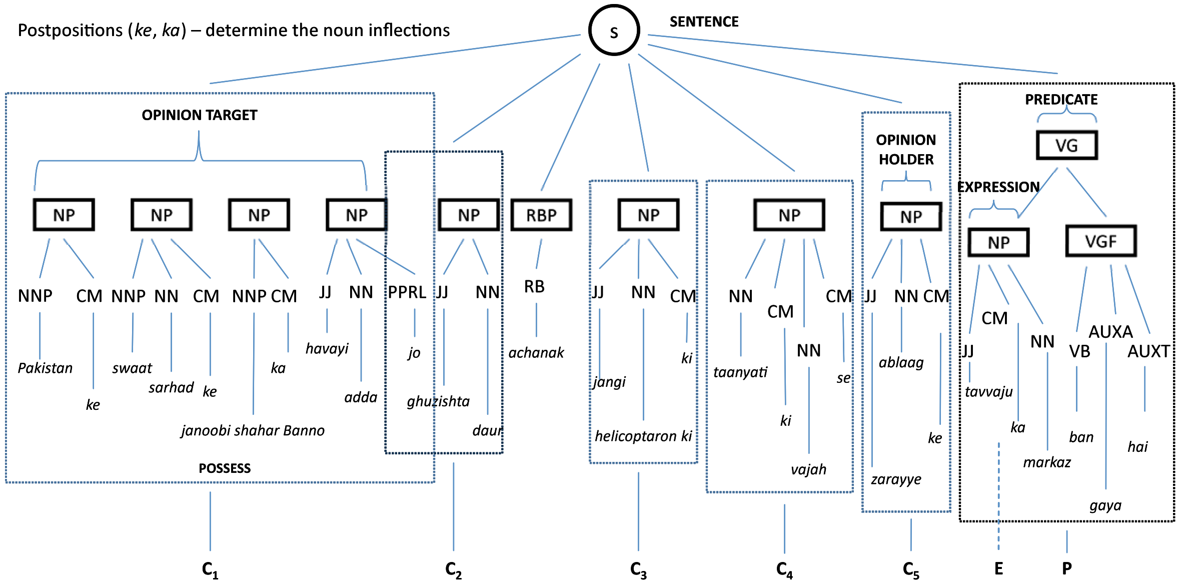
There are three advantages in our approach to creating candidate sequences: (i) by pairing expressions with their nearest predicates, several unnecessary candidate sequences are eliminated, (ii) phrases that do not contain nouns are automatically not considered (see RBP chunk in figure 1), and (iii) by considering only one candidate chunk at a time in generating the candidate sequence, we ensure that the sequence that is generated is short for better sequence kernel performance.

4.4.1 Linear Kernel features

For linear kernels we define features explicitly based on the lexical relationship between the candidate and its context. Table 5 outlines the features used.

Feature Sets and Description	
Set 1 **Baseline**	1. head word of candidate 2. case marker contained within candidate? 3. expression words 4. head word of predicate 5. POS sequence of predicate words 6. # of NPs between candidate and emotion
Set 2	7. the DOER 8. expression right after candidate?
Set 3	9. gender match between candidate and predicate 10. predicate contains emotion words?
Set 4	11. POS sequence of candidate
Set 5	12. *"kah"* feature in the predicate 13. locative feature? 14. genitive feature on noun?

Table 5: Linear Kernel Features


Postpositions (*ke*, *ka*) – determine the noun inflections
S
SENTENCE
OPINION TARGET
POSSESS
OPINION HOLDER
PREDICATE
EXPRESSION
NP
RBP
VG
VGF
NNP CM NNP NN CM NNP CM JJ NN PPRL JJ NN RB JJ NN CM NN CM NN CM JJ NN CM JJ CM NN VB AUXA AUXT
Pakistan ke swaat sarhad ke janoobi shahar Banno ka havayi adda jo ghuzishta daur achanak jangi helicoptaron ki taanyati ki vajah se ablaag ke zarayye tavvaju ka markaz ban gaya hai
C_1
C_2
C_3
C_4
C_5
E
P

English Translation - **Pakistan's provincial border's south city's airbase** has become the center of attraction for all **reporters** (ex. 1 in table 1)

Candidate sequence formation: **< [Candidate (*C*$_i$)], [Expression (*E*)], [Predicate (*P*)] >**

[Pakistan ke swaat sarhad ke janoobi shahar Banno ka havayi adda jo] [tavvaju ka markaz] [tavvaju ka markaz ban gaya hai] ...***opinion target***

1. [ghuzishta daur] [tavvaju ka markaz] [ban gaya hai]
2. [jangi helicoptaron ki] [tavvaju ka markaz] [tavvaju ka markaz ban gaya hai]
3. [taanyati ki vajah se] [tavvaju ka markaz] [tavvaju ka markaz ban gaya hai]
4. [zarayye ablaag ke] [tavvaju ka markaz] [tavvaju ka markaz ban gaya hai]...***opinion holder***

Figure 1: Illustration of candidate sequences

4.4.1 Sequence Kernel features

Features commonly used for sequence kernels are based on words (such as character-based or word-based sequence kernels). In this work, we consider Σ_i to be a feature space over Urdu words along with other disjoint features such as POS, gender, case inflections. In the kernel, however, for each combination (see table 6) the similarity matching function $f(s,t)$ that computes the number of similar features remains the same.

KID	Kernel Type
1	word based kernel (baseline)
2	word + POS (parts of speech)
3	word + POS + chunk
4	word + POS + chunk + gender inflection

Table 6: Disjoint feature set for sequence kernels

Sequence kernels are robust and can deal with complex structures. There are several overlapping features between the feature sets used for linear kernel and sequence kernel. Consider the POS path information feature. This is an important feature for the linear kernel. However this feature need not be explicitly mentioned for the sequence kernel as the model internally learns the path information. In addition, several Boolean features explicitly described for the linear kernel (2 and 13 in table 5) are also learned automatically in the sequence kernel by matching subsequences.

5 Experiments

The data used for our experiments is explained in §4.1. Figure 2 gives a flow diagram of the entire process. LIBSVM's (Chang and Lin, 2001) linear kernel is trained using the manually coded features mentioned in table 5. We integrated our proposed sequence kernel with the same toolkit. This sequence kernel uses the features mentioned in table 6 and the decay factor λ is set to 0.5.

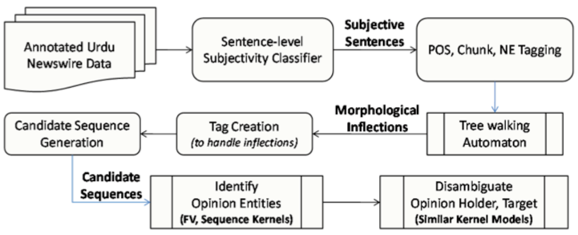

Annotated Urdu Newswire Data
Sentence-level Subjectivity Classifier
Subjective Sentences
POS, Chunk, NE Tagging
Tree walking Automaton
Morphological Inflections
Tag Creation (to handle inflections)
Candidate Sequence Generation
Candidate Sequences
Identify Opinion Entities (FV, Sequence Kernels)
Disambiguate Opinion Holder, Target (Similar Kernel Models)

Figure 2: Overall Process

The candidate sequence generation algorithm generated 8,329 candidate sequences (contains all opinion holders and targets – table 3) that are used for training both the kernels. The data is parsed using Semantex^{TM} to apply POS, chunk and morphology information. Our evaluation is based on the exact candidate boundary (whether the candidate is enclosed in a POSSESS, DOER or NP chunk).All scores are averaged over a 5-fold cross validation set.

5.1 Comparison of Kernels

We apply both linear kernels (LK) and sequence kernels (SK) to identify the entities as well as disambiguate between the opinion holders and targets. Table 7 illustrates the baselines and the best results for boundary detection of opinion entities. ID 1 of table 7 represents the result of using LK with feature set 1 (table 5). We interpret this as our baseline result. The best F1 score for this classifier is 50.17%.

ID	Kernel	Features (table 5/6)	Prec. (%)	Rec. (%)	F1 (%)
1	LK	Baseline (Set 1)	39.58	51.49	44.75
2	LK(best)	Set 1, 2, 3, 4, 5	44.20	57.99	50.17
3	SK	Baseline (KID 1)	58.22	42.75	49.30
4	SK (best)	KID 4	54.00	62.79	58.06
5	Best LK + best SK	KID 4, Set 1, 2, 3, 4, 5	**58.43**	**65.04**	**61.55**

Table 7: Boundary detection of Opinion Entities

Table 8 compares various kernels and combinations. Set 1 of table 8 shows the relative effect of feature sets for LK and how each set contributes to detecting opinion entity boundaries. Although several features are inspired by similar classification techniques (features used for SRL and opinion mining by Choi *et al.*, (2005) ~ set 1, table 5), the free word nature of Urdu language renders these features futile. Moreover, due to larger average length of each sentence and high occurrences of NPs (candidates) in each sentence, the number of candidate instances (our algorithm creates 10 sequences per sentence on average) is also very high as compared to any English corpus. This makes the training corpus highly imbalanced. Interestingly, when features like – *occurrence of postpositions, "kah" predicate, gender inflections etc.* are used, classification improves (set 1, Feature set 1,2,3,4,5, table 8).

Set	Kernel	KID	Prec. (%)	Rec. (%)	F1 (%)
1	LK	Baseline (Set 1)	39.58	51.49	44.75
		Set 1,2	39.91	52.57	45.38
		Set 1, 2, 3	43.55	57.72	49.65
		Set 1,2,3,4	44.10	56.90	49.68
		Feature set 1,2,3,4,5	**44.20**	**57.99**	**50.17**
2	SK	Baseline - KID 1	58.22	42.75	49.30
		KID 2	**58.98**	47.55	52.65
		KID 3	58.18	49.62	53.59
		KID 4	54.00	**62.79**	**58.06**
3	SK + LK	KID 1 + best LK	51.44	**68.89**	58.90
		KID 2 + best LK	**59.18**	62.98	61.02
		KID 3 + best LK	55.18	68.38	61.07
		KID 4 + best LK	58.43	65.04	**61.55**

Table 8: Kernel Performance

ID 3 of table 7 displays the baseline result for SK. Interestingly enough, the baseline F1 for SK is very close to the best LK performance. This shows the robustness of SK and its capability to learn complex substructures with only words. A sequence kernel considers all possible subsequence matching and therefore implements a concept of partial (fuzzy) matching. Because of its tendency to learn all fuzzy matches while penalizing the gaps between words intelligently, the performance of SK in general has better recall (Wang, 2008). To explain the recall situation, consider set 2 of table 8. This illustrates the effect of disjoint feature scopes of each feature (POS, chunk, gender). Each feature adds up and expands the feature space of sequence kernel and allows fuzzy matching thereby improving the recall. Hence KID 4 has almost 20% recall gain over the baseline (SK baseline). However, in many cases, this fuzzy matching accumulates in wrong classification and lowers precision. A fairly straightforward approach to overcome this problem is to employ a high precision kernel in addition to sequence kernel. Another limitation of SK is its inability to capture complex grammatical structure and dependencies making it highly dependent on only the order of the string sequence that is supplied.

We also combine the similarity scores of SK and LK to obtain the benefits of both kernels. This permits SK to expand the feature space by naturally adding structural features (POS, chunk) resulting in high recall. At the same time, LK with strict features (such as the use of *"kah"* verb) or rigid word orders (several Boolean features) will help maintain acceptable precision. By summing the contribution of both kernels, we achieve an F1 of 61.55% (Set 3, table 8), which is 17.8%, more (relative gain – around 40%) than the LK baseline results (ID 1, table 7).

Kernel	Opinion Entity	Prec. (%)	Rec. (%)	F1 (%)
LK (best)	Holder	58.71	66.67	62.44
	Target	**65.53**	57.48	61.23
SK	Holder	60.26	69.46	64.54
	Target	59.75	49.73	54.28
Both kernels	Holder	62.90	**69.81**	**65.26**
	Target	60.71	55.44	57.96

Table 9: Opinion entity disambiguation for best features

Our next sets of experiments are conducted to disambiguate opinion holders and targets. A large number of candidate sequences that are created are not candidates for opinion entities. This results in a huge imbalance in the data set. Jointly classify opinion holders, opinion targets and false candidates with one model can be attempted if this imbalance in the data set due to false candidates can be reduced. However, this has not been attempted in this work. In order to showcase the feasibility of our method, we train our model only on the gold standard candidate sequences that contain opinion entities for entity disambiguation.

The two kernels are applied on just the two classes (opinion holder vs. opinion target). Combined kernels identify holders with a 65.26% F1 (table 9). However, LK performs best for target identification (61.23%). We believe that this is due to opinion holders and targets sharing similar syntactic structures. Hence, the sequence information that SK learns affects accuracy but improves recall.

6 Challenges

Based on the error analysis, we observe some common mistakes and provide some examples.

1. Mistakes resulting due to POS tagger and shallow chunker errors.
2. Errors due to heuristic rules for morphological analysis.
3. Mistakes due to inaccurate identification of expression words by the subjectivity classifier.
4. Errors due to complex and unusual sentence structures which the kernels failed to capture.

Example (3):

Is ***na-insaafi ka badla*** **hamein** zaroor layna chahiye.
[**we** have to certainly take ***revenge for this injustice.***]

Example (4):

Kya ***hum dayshadgardi*** ka shikar banna chahatein hai?
[Do **we** want to become victims of **terrorism**?]

Example (5):

Jab ***secretary*** kisi aur say baat karke husthi hai, tho ***Pinto*** ko ghussa aata hai.
[When the ***secretary*** talks to someone and laughs, **Pinto** gets angry.]

Example 3 is a false positive. The emotion is "anger", indicated by ***"na-insaafi ka badla"*** (*revenge for injustice)* and ***"zaroor"*** *(certainly)****.*** But only the second expression word is identified accurately. The sequence kernel model determines *na-insaafi (injustice)* to be the opinion holder when it is actually the reason for the emotion. However, it also identifies the correct opinion holder - *hamein (we)*. Emotions associated with interrogative sentences are not marked (example 4) as there exists no one word that captures the overall emotion. However, the subjectivity classifier identifies such sentences as subjective candidates. This results in false negatives for opinion entity detection. The target (*secretary*) in example 5, fails to be detected as no candidate sequence that we generate indicates the noun *"secretary"* to be the target. We propose to address these issues in our future work.

7 Conclusion

We describe an approach to identify opinion entities in Urdu using a combination of kernels. To the best of our knowledge this is the first attempt where such an approach is used to identify opinion entities in a language lacking the availability of resources for automatic text processing. The performance for this task for Urdu is equivalent to the state of the art performance for English (Weigand and Klakow, 2010) on the same task.

References

Collin F. Baker, Charles J. Fillmore, John B. Lowe. 1998. The Berkeley FrameNet Project, Proceedings of the 17th international conference on Computational linguistics, August 10-14. Montreal, Quebec, Canada

Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. 2004. Automatic Extraction of Opinion Propositions and their Holders, AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications.

Kenneth Bloom, Sterling Stein, and Shlomo Argamon. 2007. Appraisal Extraction for News Opinion Analysis at NTCIR-6. In Proceedings of NTCIR-6 Workshop Meeting, Tokyo, Japan.

R. C. Bunescu and R. J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In Proceedings of HLT/EMNLP.

R. C. Bunescu and R. J. Mooney. 2005. Subsequence Kernels for Relation Extraction. NIPS. Vancouver. December.

Chih-Chung Chang and Chih-Jen Lin. 2001. LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP), Vancouver, Canada.

Aaron Culotta and Jeffery Sorensen. 2004. Dependency tree kernels for relation extraction. In Proceedings of the 42rd Annual Meeting of the Association for Computational Linguistics. pp. 423-429.

Alice Davison. 1999. Syntax and Morphology in Hindi and Urdu: A Lexical Resource. University of Iowa.

Andrea Esuli and Fabrizio Sebastiani. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In Proc of LREC. Vol 6, pp 417-422.

Scott Gimm. 2007. Subject Marking in Hindi/Urdu: A Study in Case and Agency. ESSLLI Student Session. Malaga, Spain.

Youngho Kim, Seaongchan Kim and Sun-Hyon Myaeng. 2008. Extracting Topic-related Opinions and their Targets in NTCIR-7. In Proceedings of the 7th NTCIR Workshop Meeting. Tokyo. Japan.

John Lafferty, Andrew McCallum and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA . pp. 282–289

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, Chris Watkins. 2002. Text classification using string kernels. J. Mach. Learn. Res. 2 (March 2002), 419-44.

Kim, Soo-Min. and Eduard Hovy. 2006. Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In ACL Workshop on Sentiment and Subjectivity in Text.

Alessandro Moschitti, Daniele Pighin, Roberto Basili. 2008. Tree kernels for semantic role labeling. Computational Linguistics. Vol 34, num 2, pp 193-224.

Smruthi Mukund and Rohini K. Srihari. 2010. A Vector Space Model for Subjectivity Classification in Urdu aided by Co-Training, In Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China.

Smruthi Mukund, Rohini K. Srihari and Erik Peterson. 2010. An Information Extraction System for Urdu – A Resource Poor Language. Special Issue on Information Retrieval for Indian Languages.

Ellen Riloff, Janyce Wiebe and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-03).

Rohini K. Srihari, W. Li, C. Niu, and T. Cornell. 2008. InfoXtract: A Customizable Intermediate Level Information Extraction Engine, Journal of Natural Language Engineering, Cambridge U. Press, 14(1), pp. 33-69.

Veselin Stoyanov and Claire Cardie. 2008. Annotating Topic Opinions. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Morocco.

John Shawe-Taylor and Nello Cristianni. 2004. Kernel methods for pattern analysis. Cambridge University Press.

Mengqiu Wang. 2008. A Re-examination of Dependency Path Kernels for Relation Extraction, In Proceedings of IJCNLP 2008.

Michael Wiegand and Dietrich Klalow. 2010. Convolution kernels for opinion holder extraction. In Proc. of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. pp 795-803, ACL

Vladimir Vapnik, S.Kotz. 2006. Estimation of Dependences Based on Empirical Data. Springer, 510 pages.