Interactive Part-of-Speech Exploration

Abstract

We discuss the design of a tool for the interactive exploration of part-of-speech classes using structural features. At the heart of the tool are incremental hierarchical clustering algorithms. The algorithms are used to detect classes using morphological and syntactical features. The algorithms have been modified or designed to allow interactive exploration and constrained clustering. We present preliminary results for a corpus in the Indonesian language that show the performance and illustrate the potential of our approach.

1 Introduction

Part-of-speech tagging is the task of assigning the correct class (part-of-speech, word class or lexical category, loosely speaking) to each word in a sentence.

Classes are defined and recognized by means of structural (morphological and syntactical) and semantic criteria. Classes and criteria, while relatively well understood for most Western languages whose grammarians have been busy studying and passionately debating them since classical antiquity, remain an important and fundamental topic of research for other less studied languages.

This is particularly the case for languages of the Austronesian family like the dynamic Malay and Indonesian languages with dialects and usages that arguably fall into the group Flexible-Syntactic-Category languages (Gil, 2007).

Clustering is the task of grouping objects according to their features so that objects within the same cluster are similar and objects belonging to different clusters are dissimilar. Clustering determines intrinsic classes in a set of unlabeled objects provided relevant features and metrics.

In this paper we are discussing the design of a tool for the interactive exploration of part-of-speech classes using structural features. At the heart of the tool are incremental hierarchical clustering algorithms. The algorithms are used to detect classes using structural features such as morphological and syntactical ones. The algorithms have been modified or designed to allow interactive exploration (doing and undoing clusters) and constrained clustering (preventing or forcing objects and clusters to merge by means of constraints).

In Section 2 we briefly outline the main references in part-of-speech tagging automatic part of speech tagging, as well as we mention the previous work in automatic part-of-speech tagging. In Section 3, we describe the two clustering algorithms used: Single-link Agglomerative Hierarchical Clustering and Borůvka Hierarchical Clustering. While the former is well known, the latter is our original design. We discuss in Section 4 the structural features that can be used and how the two algorithms can be adapted to provide an interactive discovery tool. In Section 5, we present preliminary results for a corpus in the Indonesian language that show the performance and illustrate the potential of our approach. Finally, we conclude in Section 6.

2 Related Works

The simplest part-of-speech taggers are based on n-gram models (Charniak et. al., 1993), where a word is assigned a tag that has the highest conditional probability of occurring together with the preceding n-1 words and their respective tags. N-gram taggers require relatively large tagged training data. Transformation-based tagging (Brill, 1993) is an example of rule-based machine learning that

learns the rules of tagging from a large set of tagged training data. Unlike n-gram or transformation-based tagging, Hidden Markov Models (Cutting et. al., 1991) do not require labeled training data but require a lexicon that specifies possible part-of-speech tags for every word.

Schutze (1999) proposes the first algorithm for tagging words whose part-of-speech properties are unknown. Similarity between two words is first determined using their left and right neighbors. Each word is represented by a feature vector with one dimension for each neighbor; the cosine between these feature vectors determines the similarity between the corresponding words. Using this similarity measure, words are clustered using Buckshot algorithm (Cutting et. al., 1992) that first employs hierarchical clustering algorithm to find centroids and then uses these centroids as initial centroids for k-means clustering.

In (Bressan et. al., 2004), the authors extend Schutze's approach by considering a broader context for feature vectors. This approach was shown to be superior over all other existing methods with the Brown corpus. In this paper we will be using the extended Schutze's feature vectors. In (Bressan et. al., 2004), the authors also observe, on an Indonesian language corpus, that words with the same affixes tend to be in the same cluster, thus confirming the potential role of morphology and its interaction with syntax in the definition of part-of-speech classes.

3 Proposed Methods

We group words into parts-of-speech classes based on the cosine similarity of their feature vectors. The problem is one of clustering in a dense graph whose vertices are words and edges are weighted by similarity.

We evaluate two hierarchical clustering algorithms for this purpose. Hierarchical clustering is chosen because it does not need the number of clusters to be provided a priori and because the resulting hierarchy of clusters provides a chance for user interactivity in-between the hierarchy levels. The two hierarchical clustering algorithms evaluated are single-link agglomerative hierarchical clustering (Cutting et. al., 1992) and our own Borůvka hierarchical clustering.

3.1 Single-link Agglomerative Hierarchical Clustering

Single-link agglomerative hierarchical clustering treats each vertex as a separate cluster initially. It then scans through the list of edges (from heaviest to lightest), and iteratively merges pairs of clusters connected by the heaviest edge until there is only one cluster left. Single-linkage agglomerative clustering is essentially Kruskal's algorithm (1956) for finding a maximum spanning tree in the edge-weighted graph.

The pseudocode for single-link agglomerative hierarchical clustering is shown in figure 1.

Algorithm: Single-link Agglomerative Hierarchical Clustering ( )
Let *E* be the set of edges in the graph
- Treat each vertex as a singleton cluster at level 0
- Sort *E* from heaviest to lightest edge weights
- While the highest level cluster does not contain all vertices, take the next heaviest edge *e* from *E* (say *e* connects vertices *A* and *B*)
  - If *A* belongs to a non-singleton cluster and *B* is a singleton cluster, include *B* in *A*'s cluster. Similarly, If *B* belongs to a non-singleton cluster and *A* is a singleton cluster, include *A* in *B*'s cluster
  - Else if *A* and *B* belong to different clusters, create a new cluster at the level above the maximum level of the two clusters. Save the information of the two clusters (i.e. their levels, their members) and update their members to belong to the new cluster
- Output the clusters at each level

Figure 1. Pseudocode for Single-link Agglomerative Hierarchical Clustering

3.2 Borůvka Hierarchical Clustering

Since hierarchical clustering is essentially finding a maximum spanning tree in the edge-weighted graph, we propose a hierarchical clustering that is based on Borůvka algorithm (1926) for finding the maximum spanning tree in an edge-weighted graph. Borůvka algorithm treats each vertex as a separate cluster initially. It then scans through the list of clusters, merging each cluster to another cluster to which it is connected with its heaviest edge, until there is only one cluster left.

The pseudocode for Borůvka hierarchical clustering is shown in figure 2.

Algorithm: Borůvka Hierarchical Clustering ( )

- L = 0
- Treat each vertex as a singleton cluster at level L
- While level L contains more than one cluster
  - While there are still clusters at level L, take a cluster, say C, from this level
    - Find the lightest edge connecting C to another cluster, say D
    - If D is at level L, create a new cluster at level L+1. Save the information of C and D (i.e. their levels, their members); remove C and D from the list of clusters at level L, and update their members to belong to the new cluster
    - Else if D is a cluster at level higher than L, save the information of C (i.e. its level, its members); remove C from the list of clusters at level L, and update C's members to belong to D
  - L++
- Output the clusters at each level

Figure 2. Pseudocode for Borůvka Hierarchical Clustering

4 A Tool for Interactive POS Exploration

4.1 Feature Vectors

Using the extended Schutze's feature vectors approach, we measure similarity of words by the degree to which they share the same two neighbors on the left and on the right, respectively.

The counts of neighbors are assembled into a vector, with one dimension for each neighbor. Each word is represented by four feature vectors: left vector (corresponding to the word's immediate left neighbor), right vector (corresponding to the word's immediate right neighbor), secondary left vector (corresponding to neighbor that precedes the word's immediate left neighbor), secondary right vector (corresponding to neighbor that follows the word's immediate right neighbor). For example, if w1, w2, w3, w4 are neighbors to be considered, the word w – assuming it only occurs in the phrase (w1 w2 w w3 w4) – is represented by its left vector: (0 1 0 0), its right vector: (0 0 1 0), its secondary left vector: (1 0 0 0), and its secondary right vector: (0 0 0 1). In our approach, each feature vector consists of 3000 entries, corresponding to the 3000 most frequent words in the corpus. Each word is therefore represented by a vector of 12000 entries. Similarity between words is measured as cosine of their representative vectors.

Since Indonesian language is rich in derivational morphology that often indicates parts-of-speech, in future we can also incorporate morphological features into the vector. For example, having an entry in the vector that is set to 1 if the word has a certain affix (e.g. "pe-" which often indicates a noun) and is set to 0 if otherwise.

4.2 Interactive Clustering

Borůvka hierarchical clustering forms clusters by levels (cf. Figure 2). Because of this property, Borůvka hierarchical clustering algorithm can incorporate users' interactivity in-between levels quite easily.

After processing each level, resulting clusters can be displayed and user can be asked to input his constraints: which clusters to be broken and which clusters to be merged. The clusters are then refreshed (broken or merged) to satisfy the constraints and the process repeats.

After the user agrees to the clustering at that level, the clusters at the next level can be formed and displayed.

4.3 Constrained Clustering

At each level of the hierarchy of clusters, the clusters are displayed and user can be asked to input his constraints. The constraints can be in the form of words or morphological constraints.

Words constraints dictate which words to exclude from one another (exclusion list) and which words to include to one another (inclusion list).

Morphological constraints can be added so that words with the same affixes are grouped in the same cluster. For example words with affix "me-" that often indicates verbs, must be included with words with affix "di-" that often also indicates verbs.

The clusters are then refreshed to reflect the constraints and the process repeats.

5 Experiments

5.1 Experimental Setup

We evaluate the proposed methods using the Indonesian language corpus used in (Jelita Asian

et. al., 2004). The corpus is made of 3000 sentences; each sentence is a document from Indonesian online newspaper Kompas, dated from January – June 2002 inclusive.

We obtain 3000 most frequent words in the corpus to compose the feature vectors. Out of these 3000 words, we select 198 words to be clustered. Since no pretagged corpus is available for the Indonesian language, we manually tag these 198 words using tags inspired by Penn Treebank tag set (Marcus et. al., 1994). The words and their tags are listed in the appendix.

We study recall, r, precision, p, and F1 measure; F1 = (2 * p * r) / (p + r).

For each hierarchy level, for each part-of-speech, we return the cluster which "best" approximates the part-of-speech, i.e. each part-of-speech is mapped to the cluster at that level which produces maximum F1-measure with respect to the part-of-speech:

$$\text{part-of-speech}(i) = \max_j \{F1(i, j)\}$$

where F1 (i, j) is the F1 measure of the cluster number j with respect to the parts-of-speech i.

The weighted average of F1 measure for each hierarchy level is calculated as:

$$F1 = \Sigma (n_i/S) * F1(i, \text{part-of-speech}(i))$$

for $0 \leq i \leq T$; where T is the number of parts-of-speech; n_i is the number of words belonging to parts-of-speech i; and S is the number of words (i.e. S = 198).

5.2 Experimental Results

We present at each hierarchy level, the weighted-average of precision, recall, and F1-measure produced by each clustering algorithm. We also present results after incorporating users' interactivity.

In figure 3 we compare the precision, recall and F1 produced at different levels by single-link and Borůvka hierarchical clustering.

From figure 3, we can see that in terms of F1 and recall, Borůvka always gives higher F1 and recall than single-link at different levels of clustering.

Since Borůvka achieves higher F1 than single-link and allows for ease of users' interactivity, for the remainder of this paper we will present Borůvka's results.

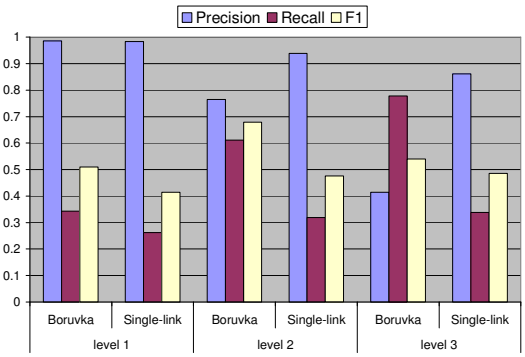

Figure 3. Borůvka and Single-link Comparison

In figure 4, we compare the precision, recall and F1 produced by Borůvka at different levels of hierarchy.

From figure 4, we can see that precision is highest at level 1 and drops at subsequent levels. Recall is lowest at level 1 and increases at subsequent levels. F1 is highest at level 2.

Precision is high at the lowest level when the clusters are small and more pure. Recall is higher at higher level because clusters are merged and bigger clusters are formed. F1, which is a harmonic average of precision and recall, is highest at level 2. This indicates that the best clustering result is formed at level 2.

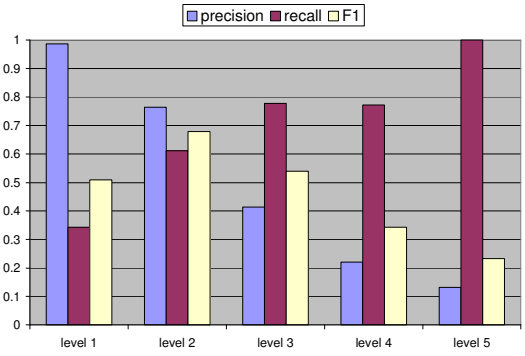

Figure 4. Borůvka at Different Levels

In figure 5, we present results when users' interactivity in the form of words' constraints (Borůvka-Word), morphological constraints (Borůvka-Morph), and both words and morphological constraints (Borůvka-Word-Morph) is added to level 1 of Borůvka hierarchical clustering.

From figure 5 we can see that adding words and morphological constraints improves both precision and recall. In particular, adding both words and morphological constraints (Borůvka-

Word-Morph) gives the highest improvement in precision, recall and F1 over the original clustering.

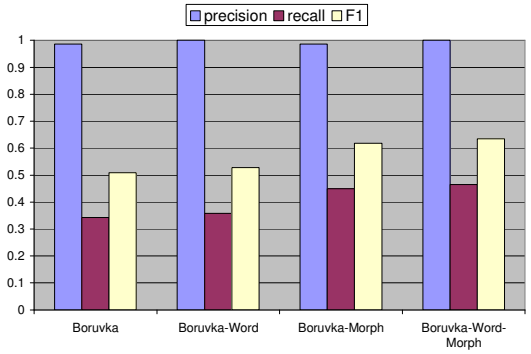

Figure 5. Adding Constraints to Borůvka (At Level 1)

In figure 6, we compare the F1 values of Borůvka and Borůvka-Word-Morph at different levels of clustering.

From figure 6, we can see that Borůvka-Word-Morph gives higher F1 than Borůvka at different levels of clustering.

In particular, Borůvka-Word-Morph gives the highest F1 at level 2 of clustering; indicating that best clustering result is indeed formed at level 2.

Although in this experiment we can achieve optimality (F1 = 1.0) at low hierarchy level (i.e. level 2), we believe the small number of words we cluster may contribute to this quickness of merging. When number of words to be clustered is large, the merging may not be so quick and optimality may be achieved only at higher level.

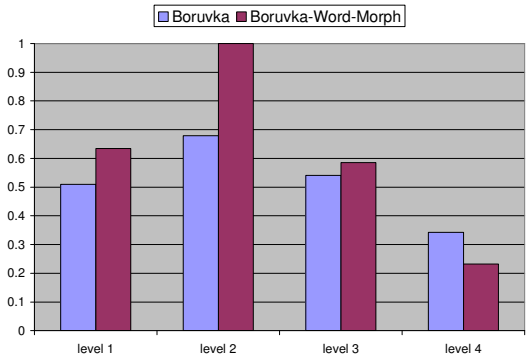

Figure 6. F1 Values after Adding Constraints to Borůvka (At All Levels)

5.3 Semantic Senses and Name-Entity Recognition

We observe interesting findings in our experiment that is consistent with the previous findings in (Bressan et. al., 2004) and suggests semantic significance to our results beyond parts-of-speech tagging.

In particular our findings relate to word senses and name-entity recognition.

In terms of name-entity-recognition, for example, for proper nouns we achieve finer granularity of clustering at level 1 and level 2, in which names of days, months, years, places, people are grouped at different clusters (cf. Figure 7).

Tag	Examples	Note
NNP	senin selasa rabu kamis jumat sabtu minggu	days
NNP	januari februari maret april mei juni juli agustus september oktober november desember	months
NNP	1980 1998 2002 2000 2001	years
NNP	padang medan surabaya jakarta ambon italia belanda jerman cina swiss jepang as singapura timtim australia malaysia pengadilan kejaksaan	places
NNP	muzadi wahid sidiq mz bisri	last name
NNP	hasyim asyawadi cholil abdurahman nur zainuddin	first name

Figure 7. Examples of Name-entity Observed

Being rich in morphological features, Indonesian grammar can often present an array of inconsistencies and exceptions (Sneddon, 2004). For example, although the affix “ter-” is often used to create adjectives (e.g. terkenal : famous); some base words (e.g. sangka, sebut) when combined with the affix “ter-” can produce a noun (i.e. tersangka : suspect) or a determinant (i.e. tersebut : the) instead of an adjective. Another example is the word pengadilan : court that has the affix “pe-an” commonly used to create nouns. However the word pengadilan is more commonly used in conversation and sentences to refer to a place instead of a noun.

Another example is the repeat words in Indonesian language (e.g. orang-orang : people) that are often used as nouns. However, some base words (e.g. hati, pelan) when repeated produce adjectives (i.e. hati-hati : careful, pelan-pelan : slow) instead of nouns.

Faced with these inconsistencies, our clustering method is able to find the correct sense of the word, therefore the correct grouping instead of being sidetracked by the morphological/grammar features (cf. Figure 8).

Tag	Examples	Note
DT	ini itu tersebut	this, that, the
JJ	pelan-pelan terbuka lambat hati-hati cepat	
NN	tokoh-tokoh orang-orang	
NNP	padang kejaksaan pengadilan medan surabaya jakarta ambon	places
NN	tersangka saksi	

Figure 8. Examples of Word Senses Observed

6 Conclusion

We have presented a tool for the interactive and constrained exploration of part-of-speech classes using structural features. The tool relies on incremental hierarchical clustering algorithms. Our preliminary results for a corpus in the Indonesian language show that the performance is satisfactory even for a small set of words.

References

Jelita Asian, Hugh Williams, and Seyed Tahaghoghi. 2004. *A Testbed for Indonesian Text Retrieval*. In Proceedings of the 9th Australasian Document Computing Symposium, Melbourne, Australia : 55-58.

Otakar Borůvka. 1926. *O Jistém Problému Minimálním (About a Certain Minimal Problem)*. Práce mor. přírodověd. spol. v Brně III 3: 37–58.

Stéphane Bressan and Lily Indrajaja. 2004. *Part-of-speech Tagging without Training*. In Proceedings of IFIP International Conference, INTELLCOMM 2004, Bangkok, Thailand.

Eric Brill. 1993. *Automatic Grammar Induction and Parsing Free Text: A Transformation-based Approach*. In Proceedings of ACL 31, Columbus, OH.

Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowitz. 1993. *Equation for Part-of-speech Tagging*. In proceedings of the Eleventh National Conference on Artificial Intelligence: 784 - 789.

Douglas Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1991. *A Practical Part-of-speech Tagger*. In the 3rd Conference on Applied Natural Language Processing, Trento, Italy.

Douglas Cutting, Jan Pedersen, David Karger, and John Tukey. 1992. *Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections*. In Proceedings of SIGIR ‘g2: 318-329.

David Gil. 2007. *On the position of Riau Indonesian in a typology of "Flexible-Syntactic-Category" languages*. Oral communication in Workshop on Languages with Flexible Parts-of-Speech Systems, University of Amsterdam. (http://www.fgw.uva.nl/aclc)

Joseph Kruskal. 1956. *On the Shortest Spanning Subtree and the Traveling Salesman Problem*. In Proceedings of the American Mathematical Society 7: 48–50.

Mitchell Marcus, Grace Kim, Mary Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. *The Penn Treebank: Annotating Predicate Argument Structure*. In ARPA Human Language Technology Workshop.

Hinrich Schutze. 1999. *Distributional Part-of-speech Tagging*. In EACL7: 141-148.

James Sneddon. 2004. *The Indonesian Language: Its History and Role in Modern Society*. USNW Press.

Appendix. List of Words and Tags

Tag	Words
Verb	masuk bernilai beranggotakan berpendapat bertahan mengikuti menghadiri menghadapi mencapai memberikan mewujudkan mencari menilai menyebutkan mengatakan membuktikan mengakui disampaikan ditahan didampingi dikuasai ditinggali dipimpin dihubungi ditanya dinilai diduga
PRP$	lanjutnya tegasnya tambahnya katanya ujarnya
WRP	apa siapa
WRB	kapan bagaimana kenapa mengapa
CD	satu dua tiga empat lima enam
DT	ini itu tersebut setiap beberapa sebuah seorang seekor
CC	sedangkan termasuk karena seperti dimana jadi meski namun tapi lewat melalui berdasarkan yang dan sehingga
JJ	pelan-pelan terbuka merah kuning terakhir ketiga pertama kedua lambat hati-hati cepat ungu biru hitam
PRP	ia dia kamu beliau anda kalian dirinya aku saya mereka kita kami
RB	jangan tidak belum mampu sedikit boleh mungkin bisa perlu mulai cukup agak sangat keras sulit mudah mesti akan ingin semua banyak ada pernah
NN	gol mobil lembar ekor buah orang terdakwa kemenangan nilai tersangka saksi tokoh-tokoh orang-orang kebijakan kejaksaan keputusan pernyataan pejabat pemain pertandingan pertemuan pengadilan penduduk
IN	tentang mengenai di ke oleh pada dari dengan
NNP	padang medan surabaya jakarta ambon italia belanda jerman cina swiss jepang muzadi wahid sidiq mz bisri september desember juni mei april februari januari maret selasa rabu as singapura oktober agustus juli november minggu sabtu jumat senin kamis 1980 1998 2002 2000 2001 hasyim asyawadi abdurahman nur zainuddin cholil timtim australia malaysia