

NAACL HLT 2009

**The Third International Workshop
on Cross Lingual
Information Access:
Addressing the Information Need
of Multilingual Societies (CLIAWS3)**

Proceedings of the Workshop

**June 4, 2009
Boulder, Colorado**

Production and Manufacturing by
Omnipress Inc.
2600 Anderson Street
Madison, WI 53707
USA

©2009 The Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-932432-33-6

Introduction

The development of digital and online information repositories is creating many opportunities and also new challenges in information retrieval. The availability of online documents in many different languages makes it possible for users around the world to directly access previously unimagined sources of information. However in conventional information retrieval systems the user must enter a search query in the language of the documents in order to retrieve it. This requires that users can express their queries in those languages in which the information is available and can understand the documents returned by the retrieval process. This restriction clearly limits the amount and type of information that an individual user really has access to.

Cross Lingual Information Access is concerned with technologies that let users express their query in their native language, and irrespective of the language in which the information is available, present the information in the user-preferred language or set of languages, in a manner that satisfies the user's information needs. The additional processing may take the form of machine translation of snippets, summarization and subsequent translation of summaries and/or information extraction.

In recent times, research in Cross Lingual Information Access has been vigorously pursued through several international fora, such as, the Cross-Language Evaluation Forum (CLEF), NTCIR Asian Language Retrieval, Question-answering Workshop and such other fora. A workshop geared towards cross language information retrieval in Indian languages (FIRE) was organized in December 2008. In addition to CLIR, significant results have been obtained in multilingual summarization workshops and cross-language named entity extraction challenges by the ACL (Association for Computational Linguistics) and the Geographic Information retrieval (GeoCLEF) track of CLEF.

The previous two issues of this workshop were held in January 2007, during IJCAI 2007 in Hyderabad, India (<http://search.iiit.ac.in/CLIA2007/>) and subsequently during IJCNLP 2008 in Hyderabad, India (<http://search.iiit.ac.in/CLIA2008/>). Both the previous workshops attracted an encouraging number of submissions, and a large number of registered participants.

This third international workshop on Cross Lingual Information Access aims to bring together various trends in multi-source, cross and multilingual information retrieval and access, and provide a venue for researchers and practitioners from academia, government, and industry to interact and share a broad spectrum of ideas, views and applications. The present workshop includes an invited keynote talk, presentations of technical papers selected after peer review followed by a panel discussion.

The workshop starts with an invited keynote talk titled Cross-Language Information Access: Looking Backward, Looking Forward by Douglas W. Oard. The talk starts with a brief recapitulation of two earlier generations of automated support for cross-language information access, the first from roughly 1964 to 1985, and the second from roughly 1989 to the present. With that as background, the talk takes stock of where we are, and where we see unmet needs that call for capabilities beyond what can currently be accomplished. It will be concluded with a few observations about how we might expect the role of the research community to evolve as progressively more capable cross-language information access technologies become commercially viable. In the other paper in the first session, Zhuang et al. report a quasi-language-independent subword recognizer trained on multiple languages, to obtain an abstracted representation of speech data in an unknown language. A retrieval model based on finite

state machines for fuzzy matching of speech sound patterns, and further for speech retrieval has been proposed. A pilot study of speech retrieval in unknown languages is presented using English, Spanish and Russian as training languages, and Croatian as the unknown target language.

In the second session, Raj and Maganti present a transliteration based search engine capable of searching 10 multi-script and multi-encoded Indian languages content on the web. Bouma et al. present a method for cross-lingual alignment of template and infobox attributes in Wikipedia. Elena Filatova presents preliminary results on quantifying Wikipedia multilinguality which show that asymmetries in multilingual Wikipedia do not make it an undesirable corpus for NLP applications training.

In the third session, Zubaryeva and Savoy present a new statistical approach to opinion detection and its evaluation on the English, Chinese and Japanese corpora. Katragadda et al. describe a sentence position based summarizer based on a sentence position policy, created from the evaluation test bed of recent summarization tasks at Document Understanding Conferences (DUC). Sankar and Sobha propose an efficient text summarization technique that involves two basic operations, finding coherent chunks in the document and ranking the text in the individual coherent chunks and picking the sentences that rank above a given threshold.

In the fourth and final session of the workshop, Mukund and Srihari propose a bootstrapped model that involves four levels of text processing for Urdu and show that increasing the training data for POS learning by applying bootstrapping techniques improves NE tagging results. The workshop concludes with a panel discussion.

We thank Douglas W. Oard for the invited keynote talk, all the members of the Program Committee for their excellent and insightful reviews, the authors who submitted contributions for the workshop and the participants for making the workshop a success. We also express our thanks to Asif, Partha and Babji who helped us in organizing and preparing the proceedings as well as maintaining the workshop webpage.

Organizing Committee

The Third International Workshop on Cross Lingual Information Access

NAACL-HLT 2009

June 4, 2009.

Organizers:

Sivaji Bandyopadhyay, Jadavpur University
Pushpak Bhattacharyya, IIT Bombay
Vasudeva Varma, IIIT Hyderabad
Sudeshna Sarkar, IIT Kharagpur
A Kumaran, Microsoft Research India
Raghavendra Udupa, Microsoft Research India

Program Committee:

A Kumaran, Microsoft Research India
Asif Ekbal, Jadavpur University
Carol Peters, ISTI and CLEF Campaign
Gregory Grefenstette, Exalead, France
Mandar Mitra, ISI Kolkata
Paolo Rosso, University Politecnica de Valencia
Patrick Saint Dizier, IRIT, Universite Paul Sabatier
Paul McNamee, John Hopkins University
Pushpak Bhattacharyya, IIT Bombay
Raghavendra Udupa, Microsoft Research India
Ralf Steinberger, European Commission-Joint Research Centre
Sivaji Bandyopadhyay, Jadavpur University
Sobha, L, AU-KBC
Sudeshan Sarkar, IIT Kharagpur
Vasudeva Varma, IIIT Hyderabad

Invited Speaker:

Douglas W. Oard, College of Information Studies and Institute for Advanced Computer Studies,
University of Maryland

Table of Contents

<i>Cross-Language Information Access: Looking Backward, Looking Forward</i>	
Douglas W. Oard	1
<i>Speech Retrieval in Unknown Languages: a Pilot Study</i>	
Xiaodan Zhuang, Jui Ting Huang and Mark Hasegawa-Johnson	3
<i>Transliteration based Search Engine for Multilingual Information Access</i>	
Anand Arokia Raj and Harikrishna Maganti	12
<i>Cross-lingual Alignment and Completion of Wikipedia Templates</i>	
Gosse Bouma, Sergio Duarte and Zahurul Islam	21
<i>Directions for Exploiting Asymmetries in Multilingual Wikipedia</i>	
Elena Filatova	30
<i>Investigation in Statistical Language-Independent Approaches for Opinion Detection in English, Chinese and Japanese</i>	
Olena Zubaryeva and Jacques Savoy	38
<i>Sentence Position revisited:</i>	
<i>A robust light-weight Update Summarization ‘baseline’ Algorithm</i>	
Rahul Katragadda, Prasad Pingali and Vasudeva Varma	46
<i>An Approach to Text Summarization.</i>	
Sankar K and Sobha L	53
<i>NE Tagging for Urdu based on Bootstrap POS Learning</i>	
Smruthi Mukund and Rohini K. Srihari	61

Conference Program

Thursday, June 4, 2009

- 8:30–9:15 Coffee Service
- 9:15–10:30 Session 1
- 9:15–9:30 Inauguration
- 9:30–10:00 *Cross-Language Information Access: Looking Backward, Looking Forward*
Douglas W. Oard
- 10:00–10:30 *Speech Retrieval in Unknown Languages: a Pilot Study*
Xiaodan Zhuang, Jui Ting Huang and Mark Hasegawa-Johnson
- 10:30–11:00 Morning Break
- 11:00–12:30 Session 2
- 11:00–11:30 *Transliteration based Search Engine for Multilingual Information Access*
Anand Arokia Raj and Harikrishna Maganti
- 11:30–12:00 *Cross-lingual Alignment and Completion of Wikipedia Templates*
Gosse Bouma, Sergio Duarte and Zahurul Islam
- 12:00–12:30 *Directions for Exploiting Asymmetries in Multilingual Wikipedia*
Elena Filatova
- 12:30–14:00 Lunch Break
- 14:00–15:30 Session 3
- 14:00–14:30 *Investigation in Statistical Language-Independent Approaches for Opinion Detection in English, Chinese and Japanese*
Olena Zubaryeva and Jacques Savoy
- 14:30–15:00 *Sentence Position revisited:*
A robust light-weight Update Summarization ‘baseline’ Algorithm
Rahul Katragadda, Prasad Pingali and Vasudeva Varma

Thursday, June 4, 2009 (continued)

15:00–15:30 *An Approach to Text Summarization.*
Sankar K and Sobha L

15:30–16:00 Afternoon Break

16:00–17:30 Session 4

16:00–16:30 *NE Tagging for Urdu based on Bootstrap POS Learning*
Smruthi Mukund and Rohini K. Srihari

16:30-17:30 Pannel Discussion

Cross-Language Information Access: Looking Backward, Looking Forward

Douglas W. Oard

College of Information Studies and Institute for Advanced Computer Studies
University of Maryland, College Park, MD, USA

The problem of providing people with the information that they seek when that information happens to be in an unfamiliar language is not new. Rather, what is new is what we can do to help address that challenge. To illustrate this point, I'll start my talk with a brief recap of two earlier generations of automated support for cross-language information access, the first from roughly 1964 to 1985, and the second from roughly 1989 to the present. With that as background, I'll then take stock of where we are, and where I see unmet needs that call for capabilities beyond what can currently be accomplished. I'll conclude with a few observations about how we might expect the role of the research community to evolve as progressively more capable cross-language information access technologies become commercially viable.

language information retrieval, his recent research has focused on support for search and sense making in large collections of conversational media. Additional information is available at <http://www.glue.umd.edu/~oard/>.

About the Speaker

Douglas Oard holds joint appointments as an Associate Professor in the College of Information Studies and in the Institute for Advanced Computer Studies at the University of Maryland, College Park. He earned his Ph.D. in Electrical Engineering from the University of Maryland. Dr. Oard's research interests center around the use of emerging technologies to support information seeking by end users. One of the leading researchers on cross-language information retrieval, he has helped lead nine evaluation campaigns focused on that problem for the Text Retrieval Conference (TREC) and the Cross-Language Evaluation Forum (CLEF). In addition to his work on ranking algorithms and interaction design for cross-

Speech Retrieval in Unknown Languages: a Pilot Study*

Xiaodan Zhuang[#] Jui Ting Huang[#] Mark Hasegawa-Johnson
Beckman Institute, Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, U.S.A.
{xzhuang2, jhuang29, jhasegaw}@uiuc.edu

Abstract

Most cross-lingual speech retrieval assumes intensive knowledge about all involved languages. However, such resource may not exist for some less popular languages. Some applications call for speech retrieval in unknown languages. In this work, we leverage on a quasi-language-independent subword recognizer trained on multiple languages, to obtain an abstracted representation of speech data in an unknown language. Language-independent query expansion is achieved either by allowing a wide lattice output for an audio query, or by taking advantage of distinctive features in speech articulation to propose subwords most similar to the given subwords in a query. We propose using a retrieval model based on finite state machines for fuzzy matching of speech sound patterns, and further for speech retrieval. A pilot study of speech retrieval in unknown languages is presented, using English, Spanish and Russian as training languages, and Croatian as the unknown target language.

1 Introduction

Dramatic increase in recorded speech media calls for efficient retrieval of audio files. Accessing speech media of a foreign language is a particularly important and challenging task, often referred to as cross-lingual speech retrieval or cross-lingual spoken document retrieval.

*This research is funded by NSF grants 0534106 and 0703624. The authors would like to thank Su-Youn Yoon for inspiring discussion. [#]The student authors contribute equally.

Previous work on cross-lingual speech retrieval mostly leverages on intensive knowledge about all the languages involved. Most reported work investigates retrieval in a target language, in response to audio or text queries given in a different source language (Meng et al., 2000; Virga and Khudanpur, 2003). Usually, the speech media in the target language, and the audio queries in the source language, are converted to speech recognition transcripts using large-vocabulary automatic speech recognizers (LVASR) trained for the target language and the source language respectively. The text queries, or transcribed audio queries, are translated to the target language. Text retrieval techniques are applied to retrieve speech, by retrieving the corresponding LVASR transcription in the target language. In such systems, a large-vocabulary speech recognizer trained on the target language is essential, which requires the existence of a dictionary and labeled acoustic training data in that language.

LVASR currently do not exist for most of the 6000 languages on Earth. In some situations, knowledge about the target language is limited, and definitely not sufficient to enable training LVASR. Imagine an audio database in a target language unknown to a user, who needs to retrieve spoken content relevant to some audible query in this unknown language. For example, the user knows how the name “Obama” is pronounced in the target language, and wants to retrieve all spoken documents that contain the query word, from a database in this unknown language. A linguist might find himself/herself in this scenario when he or she tries to collect a large number of utterances containing some particular

phrases in an unknown language. Similarly, an information analyst might wish to leverage on speech retrieval in unknown languages to organize critical information before engaging linguistic experts for finer analysis. We refer to such retrieval tasks as *speech retrieval in unknown languages*, in which little knowledge about the target language is assumed.

A human linguist attempting to manually perform speech retrieval in an unknown language would necessarily map the perceived speech (both database and query) into some cognitive abstraction or schema, representing, perhaps, the phonetic distinctions that he or she has been trained to hear. Matching and retrieval of speech would then be performed based on such an abstraction. Two cognitive processes, assimilation and accommodation, take place when human brains are to process new information (Bernstein et al., 2007), such as speech in an unknown language. In accommodation, the internal stored knowledge adapts to new information with which it is confronted. In assimilation, the new information, e.g., speech in an unknown language, is mapped to previously stored information, e.g., sub-words (phones) as defined by knowledge about the languages known to the listener.

This paper models speech retrieval in unknown languages using a machine learning model of phonetic assimilation. A quasi-language-independent subword recognizer is trained to capture salient sub-words and their acoustic distribution in multiple languages. This recognizer is applied on an unknown language, therefore mapping segments of the unknown speech to subwords in the known languages. Through this machine cognitive process, the database and queries in the unknown language are represented as sequences of quasi-language-independent subwords. Speech retrieval is performed based on such representation. Figure 1 illustrates that speech retrieval in an unknown language can be modeled as a special case of assimilation.

This task differs from the more widely studied known-language speech retrieval task, in that no linguistic knowledge of the target language is assumed. We can only leverage on knowledge that can be applied by assimilation to the multiple known languages. Therefore, this task is more like a cross-lingual sound pattern retrieval task, leveraged on quasi-language-independent subwords, rather than

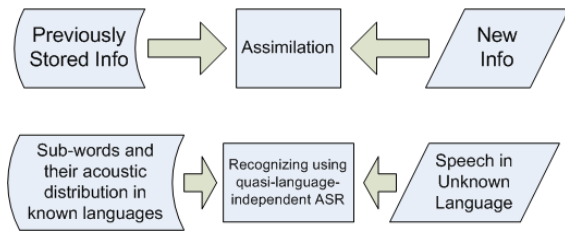


Figure 1: Automatic speech retrieval in an unknown language (below) is modeled as a special case of the cognitive process called assimilation (above).

a translated spoken word/phrase retrieval task using target language LVASR transcripts, as in most cross-lingual speech retrieval systems. The quasi-language-independent subword recognizer is trained on speech data other than the target language, and therefore generates much noisier recognition results, owing to potential mismatch between acoustic distributions, lack of dictionary and lack of a word-level language model.

To manage the extra difficulty, we adopt a subword lattice representation to encode a wide hypothesis space of recognized speech in the target language. Language-independent query expansion is achieved either by allowing a wide lattice output for an audio query, or by taking advantage of distinctive features in speech articulation to propose quasi-language-independent subwords most similar to the given subwords in a query. Finite state machines (FSM) constructed from the speech lattices are used to allow for fuzzy matching of speech sound patterns, and further for retrieval in unknown languages.

We carry out a pilot study of speech retrieval in unknown languages, using English, Spanish and Russian as training languages, and Croatian as the unknown target language. To explain the effect of additional knowledge about the target language, we demonstrate the improvements in retrieval performance that result by incrementally making available subword sequence models and acoustic models for the target language.

2 Quasi-Language-Independent subword Models

2.1 Deriving a subword set

Based on the assumption that an audible phrase in an unknown language can be represented as a sequence

of subwords, the question is to find an appropriate set of subword symbols. Schultz and Waibel (2001) reported that a global unit set for the source languages based on International Phonetic Alphabet (IPA) symbols outperforms language-dependent phonetic units in cross-lingual word recognition tasks, whereas language-dependent phonetic units are better models for multilingual word recognition (in which the target language is also one of the source languages). A multilingual task might benefit from partitioning the feature space according to language identity, i.e., to have different subsets of models aiming at different languages. By contrast, a cross-lingual task calls for one consistent set of models with language-independent properties in order to maximize portability into the new language.

To capture the necessary distinctions between different phones across languages, we first pool together individual phone inventories for source languages, each of which has its phones tagged with a language identity, and then performed bottom-up clustering on the phone pool based on pairwise similarity between their acoustic models. Each cluster represents one distinct language-independent subword symbol. Since this set is still derived from multiple languages, we refer to these subword units as *quasi-language-independent subwords*. A quasi-language-independent subword set is derived by the following steps:

First, we encode all speech in the known languages using a language-dependent phone set. Each symbol in this set is defined by the phone identity and the language identity. One single-Gaussian three-state left-to-right HMM is trained for each of these subword units.

Second, similarity between the language-dependent phones is estimated by the approximated KL divergence between corresponding acoustic models. As shown in (Vihola et al., 2002), KL divergence between single-Gaussian left-to-right HMMs can be approximated in closed form by Equation 1,

$$\begin{aligned}
 KLD(U, V) &= \sum_{i=1}^S r_i \sum_{j=1}^S a_{ij}^U \log(a_{ij}^U / a_{ij}^V) \quad (1) \\
 &+ \sum_{i=1}^S r_i I(b_i^U : b_i^V), \quad (2)
 \end{aligned}$$

where a_{ij} is the transition probability to hidden state j , and b_i and r_i are the observation distribution and steady-state probability for hidden state i . For single-Gaussian distribution, $I(b_i^U : b_i^V)$ can be approximated by,

$$\begin{aligned}
 I(b_i^U : b_i^V) &= \frac{1}{2} \left[\log \frac{|\Sigma_i^V|}{|\Sigma_i^U|} \right. \\
 &+ \text{tr} \left(\Sigma_i^U \left((\Sigma_i^V)^{-1} - (\Sigma_i^U)^{-1} \right) \right) \\
 &\left. + \text{tr} \left((\Sigma_i^V)^{-1} (\mu_i^U - \mu_i^V) (\mu_i^U - \mu_i^V)^T \right) \right].
 \end{aligned}$$

Third, we use the Affinity Propagation algorithm (Frey and Dueck, 2007) to conduct pairwise clustering of phones based on the approximated KL divergence between acoustic models. The tendency for a data point (a phone) to be an exemplar of a cluster is controlled by the preference value assigned to that phone. The preference of a phone i is set as follows to favor frequent phones to be cluster centers:

$$p(i) = k \log(C_i), \quad (3)$$

where C_i is the count of the phone i , and k is a normalization term to control the total number of clusters. To discourage subwords from the same language to join a same cluster, pairwise distance between them are offset by an additional amount, comparable to the maximum pairwise distance between the models.

The resultant subword set is supposed to capture quasi-language-independent phonetic information, and each subword unit has relatively distinctive acoustic distribution. These subwords are encoded using the corresponding cluster exemplars as surrogates.

2.2 Recognizing subwords

An automatic speech recognition (ASR) system (Jelinek, 1998) serves to recognize both queries and speech database, with acoustic models for the language-independent subwords derived from the known languages as described in section 2.1. The front-end features extracted from the speech data are 39-dimensional features including 12 Perceptual Linear Prediction (PLP) coefficients and their energy, as well as the first-order and second order regression coefficients.

We create context-dependent models for each subword, using the same strategy for building context-dependent triphone models in LVARSR (Woodland et al., 1994). A “triphone” is a subword with its context defined as its immediate preceding and following subwords. Each triphone is represented by a continuous three-state left-to-right Hidden Markov Model (HMM). Additionally, there is a one-state HMM for silence, two three-state HMMs for noise and unknown sound respectively. The number of Gaussian mixtures (9 to 21 Gaussians) is optimized according to a development set consisting of speech in the known languages. A standard tree-based state tying technique is adopted for parameter sharing between subwords with similar contexts.

The “language model” (LM), or more precisely subword sequence model, should generalize from the known languages to the unknown language. Our trial experiments showed that unigram statistics of subwords and their triphones is more transferable across languages than N-gram statistics. We also assume that infrequent triphones are less likely to be salient units that would carry the properties of the unknown language. Thus, we select the top frequent triphones and map the rest of the triphones to their center phones, forming a mixed vocabulary of frequent triphones and context-independent subwords. The frequencies of these vocabulary entries are used to estimate an unigram LM in the ASR system. Triphones in the ASR output are mapped back to its center subwords before the retrieval stage.

3 Speech Retrieval through Subword Indexing

In many cross-lingual speech retrieval systems, the speech media are processed by a large-vocabulary automatic speech recognizer (LVARSR), which has access to vocabulary, dictionary, word language model and acoustic models for the target language. With all these resources, state-of-the-art speech recognition could give reasonable hypothesized word transcript, enabling direct application of text retrieval techniques. However, this is not the case in speech retrieval in unknown languages. Moreover, without the higher level linguistic knowledge, such as a word dictionary, this task aims to *find speech patterns that sound similar*, as approximated by sequences of quasi-language-independent

subwords. Therefore, the sequential information in the hypothesized subwords is critical.

To deal with the significant noise in the subword recognition output, and to emphasize the sequential information, we use the recognizer to obtain subword lattices instead of one-best hypotheses. These lattices can be represented as weighted automata, which are compact representations of a large number of alternative subword sequences, each associated with a weight indicating the uncertainty of the data. Therefore, indexing speech in unknown language can be achieved by indexing the corresponding weighted automata with quasi-language-independent subwords associated with the state transitions.

We adopt the weighted automata indexation algorithm reported in (Allauzen et al., 2004), which is optimal for searching subword sequences, as it takes time linear in the sum of the query size and the number of speech media entries where it appears. The automata indexation algorithm also preserves the sequential information, which is crucial for this task. We leverage on two kinds of knowledge for query expansion, namely empirical phone confusion and knowledge-based phone confusion. An illustration of our speech retrieval system is presented in Figure 2. We detail the indexing approaching as well as query expansion and retrieval in this section.

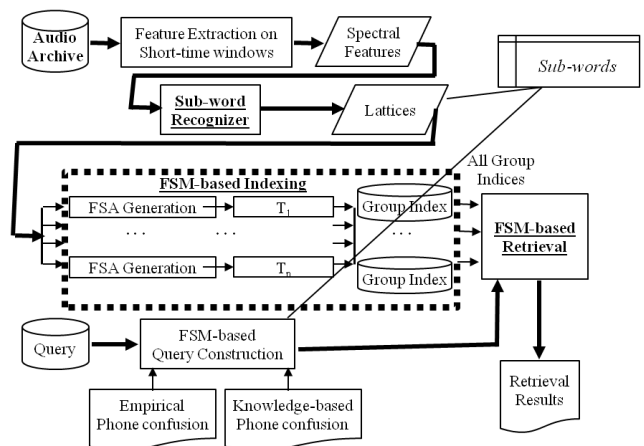


Figure 2: Framework of speech retrieval through subword indexing

3.1 Subword Finite State Machines as Speech Indices

We construct a full index that can be used to search for a query within all the speech utterances $u_i, i \in 1, \dots, n$. In particular, this is achieved by constructing a weighted finite-state transducer T , mapping each query x to the set of speech utterances where it appears. Each returned speech utterance u is assigned a score, which is the negative log of the expected count of the query x in utterance u .

The subword lattice for speech utterance u_i can be represented as a weighted finite state automata A_i , whose path weights correspond to the joint probability of the observed speech and the hypothesized subword sequence. To get an automata whose path weights correspond to desired negative log of posterior probabilities, we simply need to apply a general weight-pushing algorithm to A_i in the log semiring, resulting in an automata B_i . In this automata B_i , the probability of a given string x is the sum of the probability of all paths that contains x .

The key point of constructing the index transducer T_i for utterance u_i is to introduce new paths that enable matching between a query and any portions of the original paths, while properly normalizing the path weights. This is achieved by factor selection in (Allauzen et al., 2004). First, null output is introduced to each transition in the automata, converting the automata into a transducer. Second, a new transition is introduced from a new unique initial state to each existing state, with null input and output. The weight associated with this transition is the negative log of the forward probability. Similarly, a new transition is created from each state to a new unique final state, with null input and output as the label i of the current utterance u_i . The associated weight is the negative log of the backward probability. General finite state machine optimization operations (Allauzen et al., 2007) of weighted ϵ -removal, determinization and minimization over the log semiring can be applied to the resulting transducer. As shown in (Allauzen et al., 2004), the path with input of string x and output of label i has a weight corresponding to the negative log of the expected count of x in utterance u_i .

To optimize the retrieval time, we divide all utterances into a few groups. Within each group, the utterance index transducers are unioned and deter-

minized to get one single index transducer for the group. It is then feasible to expedite retrieval by processing each group index transducer in a parallel fashion.

3.2 Query Expansion

While sequential information is important, exact string match is very unpalatable in this challenging task, even when subword lattices encode many alternative recognition hypotheses. Language-independent query expansion is therefore critical for success in retrieval. We carry out query expansion either by allowing a wide lattice output for an audio query, or by taking advantage of distinctive features in speech articulation to propose quasi-language-independent subwords most similar to the given subwords in a query.

In particular, for a spoken query, ASR will generate a subword lattice instead of a one-best subword sequence hypothesis. With the lattice, the audio query is encoded by the best hypothesis from ASR and its empirical phone confusion. The lattice can then be represented as a finite-state automata.

However, when the query is given as a target language subword sequence, we can no longer use the recognizer to obtain an expanded query. Furthermore, some target language subwords may not even exist in the quasi-language-independent subword set in the recognizer. In this case, knowledge-based phone confusion is engaged via the use of a set of distinctive features $F_j, j \in 1, \dots, M$ for human speech (Chomsky and Halle, 1968), including labial, alveolar, post-alveolar, retroflex, voiced, aspirated, front, back, etc.

We estimate similarity from phone a to phone b , or more precisely, substitution tendency as in Equation 4,

$$DFsim(a, b) = \log \frac{N_{ab}}{N_a} \quad (4)$$

where

$$N_{ab} = \sum_{j=1}^M (F_j^a \times F_j^b \neq 1),$$

$$N_a = \sum_{j=1}^M (F_j^a \neq 0).$$

The target subword sequence is first mapped to the derived subword set, by locating the identical or nearest member phone in the clustering and then adopting the surrogate for that cluster. This converted sequence of derived subwords is further expanded by adding the most likely alternative quasi-language-independent subwords, parallel to each original subword. Transitions to these alternative subwords are associated with the corresponding substitution tendency based on distinctive features.

3.3 Search

An expanded query, either obtained from an audio query or a subword sequence query, is represented as a weighted finite state automata. Searching this query in the utterances is achieved by composing the query automata with the index transducer. This results in another finite state transducer, which is further processed by projection on output, removal of ϵ arcs and determinization. The output is a list of retrieved speech utterances, each with the expected count of the query.

Apparently, the precision and recall of the retrieval results vary with the width of the subword lattices used for indexing as well as how much the query is expanded. We control the width of the subword lattices via the number of tokens and the maximum probability decrease allowed for each step in the Viterbi decoding. The extend to which a subword sequence query is expanded is determined by the lowest allowed similarity between the original phone and an alternative phone. These parameters are set empirically.

4 Experiments

4.1 Dataset

The known language pool should cover as many language families as possible so that the derived subwords could better approximate language independence. However, as a pilot study, this paper reports experiments using only languages within the Indo-European family. Table 1 summarizes the size of speech data from each language. Croatian is used as the unknown target language, and the other three languages are the known languages used for deriving and training the quasi-language-independent subword models. We extracted 80% of all speakers

per language for training, and 10% as a development set.

Language	ID	Hours	Spks	Style
Croatian	hrv	21.3	201	Read+answers
English	hub	13.6	406	Broadcast
Spanish	spa	14.6	120	Read+answers
Russian	rus	2.5	63	Read+answers

Table 1: Summary for data: language ID, total length, number of speakers and speaking style for each language.

4.2 Settings

The speech retrieval task aims to find speech utterances that contain a particular query. We use two kinds of queries: 1) subword sequence queries, transcribed as a sequence of phonetic symbols in the target language; 2) audio queries, each being an audio segment of the speech query in the target language.

Since we aim to match speech patterns that sound like each other, the queries used in this experiment are relatively short, about 3 to 5 syllables. This adds to the challenge in that very limited redundant information is available for query-utterance matching. There are totally 40 subword sequences and 40 audio queries, each occurs in between 18 and 38 utterances out of a set of 576 utterances.

In addition to a cross-lingual retrieval system built using only the known languages, we incrementally augment resource on the target language to build more knowledgeable systems.

AMOLM0: Both the acoustic model (AM) and the language model (LM) are quasi-language-independent, trained using data in multiple known languages. This happens when no transcribed speech data or a defined phone set exist for the target language. Essentially the system has no direct knowledge about the target language.

AMOLMt: This setting examines the performance gap due to the acoustic model mismatch by using a quasi-language-independent AM, but a target language LM. Suppose that a word dictionary with phonetic transcription and possibly some text data from the target language are available, for training a target language subword LM. To find the mapping between target triphones and language-independent source AMs, linguistic knowledge and phonetic symbol notation are the only information

we can use. First, we map each of target monophones to source phone symbols: Any source cluster that contains a phonetic symbol with the same notation as the target phonetic symbol becomes a surrogate symbol for that target phone. If a target phone is unseen to the known languages, the most similar phone will be chosen first. The similarity is based on the distinctive features, as discussed in Section 3.2. Second, the target triphones are converted to possible source triphones for which acoustic models exist. Each target triphone not modeled in the source language AM is replaced with the corresponding diphone (subword pair) if it exists, otherwise the center phone.

AMtLM0: This setting examines the performance gap due to the language model mismatch by using a quasi-language-independent source LM, but a target language AM. For the source triphones and monophones that do not exist in the target AM, they are mapped to target AMs in a way similar as described above.

AMtLMt: Both AM and LM are trained for the target language. This setting provides an upper bound of the performance for different settings.

4.3 Metrics

We evaluate the performance for both subword recognition and speech retrieval, measured as follows.

Recognition Accuracy: The ground truth is encoded using subwords in the target language while the recognition output is encoded using quasi-language-independent subwords in Section 2. To measure the recognition accuracy, we label each quasi-language-independent subword cluster using the most frequent target language subword that appears in that cluster. The hypothesis subword sequence is then compared against the groundtruth using a dynamic-programming-based string alignment procedure. The recognition accuracy is defined as $REC - ACC = \frac{H-I}{N} \times 100\%$, where H , I , and N are the numbers of correct labels, insertion errors and groundtruth labels respectively.

Retrieval Precision: The retrieval performance is measured using Mean Average Precision ($IR - MAP$), defined as the mean of the Average Precision (AP) for a set of different queries x . Mean Average Precision ($IR - MAP$) can be defined in

Equation 5. n is the number of ordered retrieved utterances and R is the total number of relevant utterances. f_i is an indicator function whether the i^{th} retrieved utterance does contain the query. Precision p_m for top m retrieved utterances can be calculated as $p_m = \frac{1}{m} \sum_{k=1}^m f(k)$.

$$IR - MAP = \frac{1}{Q} \sum_{x=1}^Q AP(x),$$

$$AP(x) = \frac{1}{R(x)} \sum_{i=1}^{n(x)} f_i(x) p_i(x). \quad (5)$$

We use $IR - MAP_A$ and $IR - MAP_S$ to denote the retrieval MAP for audio queries and subword sequence queries respectively.

4.4 Results

Table 2 presents a few examples of the derived quasi-language-independent subwords. As discussed in Section 2, these subwords are obtained by bottom-up clustering of all the language-dependent IPA phones in the multiple known languages. The same IPA symbol across languages may lie in the same cluster, e.g., /z/ in Cluster 1, or different clusters, e.g., /j/ in Cluster 3 and 4. Although symbols within the same language are discouraged to be in one cluster, it still desirably happens for highly similar pairs, e.g., /i/_{rus} and /j/_{rus} in Cluster 4.

Cluster ID	Surrogate	Other phone members
1	/z/_{hub}	/z/_{spa}, /z/_{rus}, /z/_{rus}
2	/tʃ/_{rus}	/tʃ/_{hub}, /tʃ/_{spa}
3	/j/_{hub}	/j/_{spa}
4	/i/_{hub}	/i/_{rus}, /j/_{rus}

Table 2: Examples of quasi-language-independent subwords, as clusters of source language IPAs.

Table 3 compares the subword recognition and retrieval performance for the quasi-language-independent subwords and IPA phones. We can

Setting	REC - ACC	IR - MAP_A	IR - MAP_S
IPA	37.18%	17.90%	31.40%
AM0LM0	42.52%	23.24%	32.62%

Table 3: Performance of quasi-language-independent subword and IPA.

Setting	AMtLMt	AMtLMO	AMOLMt	AMOLMO
<i>REC - ACC</i>	73.45%	67.29%	49.88%	42.52%
<i>IR - MAP_A</i>	58.82%	52.38%	28.32%	23.24%
<i>IR - MAP_S</i>	76.96%	51.86%	34.95%	32.62%

Table 4: Performance of subword recognition and speech retrieval.

see that on the unknown language Croatian, the derived quasi-language-independent subwords outperform the IPA symbol set in both phone recognition and retrieval using two kinds of queries.

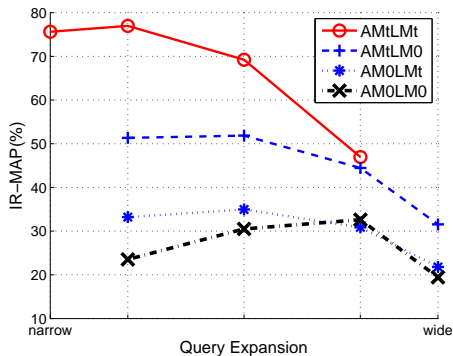


Figure 3: Speech retrieval performance for subword sequence queries

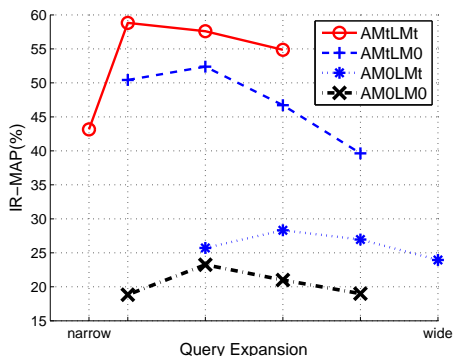


Figure 4: Speech retrieval performance for audio queries

Table 4 presents the subword recognition accuracy and retrieval performance with optimal query width. Figure 3 and Figure 4 presents speech retrieval performance at varying query widths for subword sequence queries and audio queries respectively. It is shown that speech retrieval in completely unknown language achieves MAP of 23.24% and 32.62% while the system trained using

the most available knowledge about the target language reaches MAP of 58.82% and 76.96%, for audio queries and subword sequence queries respectively. We also demonstrate access to phone frequency (AMOLMt) and acoustic data (AMtLMO) both boosts retrieval performance, and the effect is roughly additive (AMtLMt).

5 Conclusion and Discussion

In this work, we present a speech retrieval approach in unknown languages. This approach leverages on speech recognition based on quasi-language-independent subword models derived from multiple known languages, and finite state machine based fuzzy speech pattern matching and retrieval. Our experiments use Croatian as the unknown language and English, Russian and Spanish as the known languages. Results show that the derived subwords outperform the IPA symbols, and access to the subword language model and acoustic models in the unknown language explains the gap between this challenging task and retrieval with knowledge about the target language.

The proposed retrieval approach on unknown languages can be viewed as a machine learning model of phonetic assimilation, in which the segments in an unknown language are mapped to language-independent subwords learned from the multiple known languages. However, another important cognitive process, i.e., accommodation, is not yet modeled. We believe the capability to create new subwords unseen in the known languages would lead to improved performance. In particular, speech segments that are hypothesized by the quasi-language-independent subword recognizer with very low confidence scores can be clustered to form these new subwords, accommodating to the unknown language.

The approach in this work can be readily scaled up to much larger speech corpora. In particular, larger corpora would make it more practical to implement the accommodation process discussed above. Besides, that would also enable online adaptation of the model parameters of the quasi-language-independent subword recognizer. Both are believed to promise reduced gap between retrieval performance in a known language and an unknown language, and are potential future work beyond this paper.

References

- C. Allauzen, M. Mohri, and M. Saraclar. 2004. General indexation of weighted automata – application to spoken utterance retrieval. In *Proc. HLT-NAACL*.
- C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. 2007. Openfst: A general and efficient weighted finite-state transducer library. In *Proc. CIAA*.
- Bernstein, Penner, Clarke-Stewart, and Roy. 2007. *Psychology*. Houghton Mifflin Company.
- Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. New York: Harper and Row.
- Brendan J. Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315:972–976.
- Frederick Jelinek. 1998. *Statistical Methods for Speech Recognition*. The MIT Press.
- Helen Meng, Berlin Chen, Erika Grams, Sanjeev Khudanpur, Wai-Kit Lo, Gina-Anne Levow, Douglas Oard, Patrick Schone, Karen Tang, Hsin-Min Wang, and Jian Qiang Wang. 2000. Mandarin-english information (MEI): Investigating translingual speech retrieval. http://www.clsp.jhu.edu/ws2000/final_reports/mei/ws00mei.pdf.
- Tanja Schultz and Alex Waibel. 2001. Language independent and language adaptive acoustic modeling for speech recognition. *Speech Communication*, 35:31–51.
- M. Vihola, M. Harju, P. Salmela, J. Suontausta, and J. Savela. 2002. Two dissimilarity measures for hmms and their application in phoneme model clustering. In *Proc. ICASSP*, volume 1, pages I–933 – I–936.
- Paola Virga and Sanjeev Khudanpur. 2003. Transliteration of proper names in crosslingual information retrieval. In *Proc. ACL 2003 workshop MLNER*.
- P.C. Woodland, J.J. Odell, V. Valtchev, and S.J. Young. 1994. Large vocabulary continuous speech recognition using HTK. In *Proc. ICASSP*, volume 2, pages II/125–II/128.

Transliteration based Search Engine for Multilingual Information Access

Anand Arokia Raj

Speech and Language Technology Lab
Bhrigus Software (I) Pvt Ltd
Hyderabad, India
rayar.anand@bhrigus.com

Harikrishna Maganti

Speech and Language Technology Lab
Bhrigus Software (I) Pvt Ltd
Hyderabad, India
hmaganti@bhrigus.com

Abstract

Most of the Internet data for Indian languages exist in various encodings, causing difficulties in searching for the information through search engines. In the Indian scenario, majority web pages are not searchable or the intended information is not efficiently retrieved by the search engines due to the following: (1) Multiple text-encodings are used while authoring websites. (2) In spite of Indian languages sharing common phonetic nature, common words like loan words (borrowed from other languages like Sanskrit, Urdu or English), transliterated terms, pronouns etc., can not be searched across languages. (3) Finally the query input mechanism is another major problem. Most of the users hardly know how to type in their native language and prefer to access the information through English based transliteration. This paper addresses all these problems and presents a transliteration based search engine (inSearch) which is capable of searching 10 multi-script and multi-encoded Indian languages content on the web.

1 Introduction

India is a multi-language and multi-script country with 23 official languages and 11 written script forms. About a billion people in India use these languages as their first language. About 5% of the population (usually the educated class) can understand English as their second language. Hindi is spoken by about 30% (G. E. Burkhart, S. E. Goodman, A. Mehta and L. Press, 1998) of the population, but it is concentrated in urban areas and north-central India,

and is still not only foreign, but often unpopular in many other regions.

Though considerable amount of Indic content is available on the World Wide Web (WWW), we can observe that search development is very less when compared to the official languages of the United Nations (UN). The primary reason for this can be attributed for much delayed standards and lack of support from operating systems and browsers in rendering Indic scripts. This caused web publishers to develop their own proprietary encodings/fonts, who are now hesitant to use available standards such as Unicode/ISCII. This creates a major hinderance in accessing Indian content through existing search engines.

Most of the search engines support Indic search in Unicode data only. But, considerable amount of content is available in ASCII based font encodings which is much larger (more dynamic also) than Unicode (Unicode Consortium - Universal Code Standard, 1991) or ISCII (ISCII - Indian Standard Code for Information Interchange, 1983) formats. Apart from this, language independent information like loan words, transliterated words, pronouns etc., are also not accessible across Indian languages. Most users are familiar with English keyboard typing than any Indian language, and would be interested to query through English transliteration. So, a meta standard transliteration scheme (IT3 sec3.1) has to be commonly defined across all the Indian languages, and the web content has to be appropriately converted. Also, the web pages need to be indexed using phonetic features like (diphone/triphones/syllables), which will be conve-

nient to retrieve and rank the pages. In this paper, we incorporate all these aspects to make search engine as the meaningful searching tool for Indian languages.

The paper is organized into six sections. The first section explains the nature of Indic scripts. The second section details the various major encoding formats and transliteration scheme used to store and render Indic data. In section three, novel approaches for preprocessing Indic data like font-encoding identification and font-data conversion are explained. In section four, the experiments regarding stemming and grapheme-to-phoneme (G2P) for Indian-English using Classification and Regression Tree (CART) are described and stop words identification is also explained. The fifth section discusses the issues in developing a multi-lingual search engine for Indian languages. The sixth section explains the three possible ways to develop a cross-lingual search engine. Finally the report and summary are included with conclusion.

2 Nature of Indic Scripts

The scripts in Indian languages have originated from the ancient Brahmi script. The basic units of the writing system are referred to as Aksharas. The properties of Aksharas are as follows: (1) An Akshara is an orthographic representation of a speech sound (2) Aksharas are syllabic in nature (3) The typical forms of Akshara are V, CV, CCV and CCCV, thus have a generalized form of C*V. The shape of an Akshara depends on its composition of consonants and the vowel, and sequence of the consonants. In defining the shape of an Akshara, one of the consonant symbols acts as pivotal symbol (referred to as semi-full form). Depending on the context, an Akshara can have a complex shape with other consonant and vowel symbols being placed on top, below, before, after or sometimes surrounding the pivotal symbol (referred to as half-form).

Thus to render an Akshara electronically, a set of semi-full or half-forms have to be rendered, which are in turn rendered using a set of basic shapes referred to as glyphs. Often a semi-full form or half-form is rendered using two or more glyphs, thus there is no one-to-one correspondence between glyphs of a font and semi-full or half-forms.

2.1 Convergence and Divergence

All Indian languages except English and Urdu share a common phonetic base, i.e., they share a common set of speech sounds. While all of these languages share a common phonetic base, some of the languages such as Hindi, Marathi and Nepali also share a common script known as Devanagari. But languages such as Telugu, Kannada and Tamil have their own scripts. The property which distinguishes these languages can be attributed to the phonotactics in each of these languages rather than the scripts and speech sounds. Phonotactics is the permissible combination of phones that can co-occur in a language.

This knowledge helps us in designing a common transliteration scheme, and also in identifying and converting different text encodings.

3 Indic Data Formats

Another aspect involved in the diversity of electronic content of Indian languages is their format of digital storage. Storage formats like ASCII (American Standard Code for Information Interchange) based fonts, ISCII (Indian Standard code for Information Interchange), Unicode and phonetic based transliteration schemes are often used to store the digital text data in Indian languages. Most of the text is rendered using some fonts of these formats.

3.1 Phonetic Transliteration Schemes

Transliteration is a mapping from one system of writing into another, word by word, or ideally letter by letter. It is the practice of transcribing a word or text written in one writing system into another writing system. Transliterations in the narrow sense are used in situations where the original script is not available to write down a word in that script, while still high precision is required. One instance of transliteration is the use of an English computer keyboard to type in a language that uses a different alphabet, such as Russian, Hindi etc. Transliterated texts are often used in emails, blogs, and electronic correspondence where non-Latin keyboards are unavailable, is sometimes referred to by special composite terms that demonstrate the combination of English characters and the original non-Latin word pronunciation: Ruglish, Hebrish, Greeklish, Arabish or Hinlish.

To handle diversified storage formats of scripts of Indian languages such as ASCII based fonts, ISCII and Unicode etc., it is useful and becomes essential to use a meta-storage format. ISO 15919 standards (Transliteration of Indic Scripts: How to use ISO 15919,) describes development of transliteration for Indic scripts. A transliteration scheme maps the Aksharas of Indian languages onto English alphabets and it could serve as meta-storage format for text-data. Since Aksharas in Indian languages are orthographic representation of speech sound, and they have a common phonetic base, it is suggested to have a phonetic transliteration scheme such as IT3 (Ganapathiraju M., Balakrishnan M., Balakrishnan N. and Reddy R., 2005) (Prahallad Lavanya, Prahallad Kishore and GanapathiRaju Madhavi, 2005). Thus, when the font-data is converted into IT3, it essentially turns the whole effort into font-to-Akshara conversion. Thus IT3 transliteration is used as common representation scheme for all Indic data formats. The same is used to get the input from the user also.

4 Indic Data Preprocessing

In search engine development, it is an absolute requirement that the content should be in an unique format to build a efficient index table. So, preprocessing the web content is unavoidable here. Most of the Indian language electronic data is either Unicode encoded or glyph based font encoded. Processing Unicode data is quite straight forward because it follows distinguished code ranges for each language and there is a one-to-one correspondence between glyphs (shapes) and characters. But this is not true in the case of glyph based font encoded data. Hence, it becomes necessary to identify the font encoding and convert the font-data into a phonetic transliteration scheme like IT3. The following subsections explain the stages in detail.

4.1 Font-Encoding Identification

The problem of font-identification could be defined as, given a set of words or sentences to identify the font-encoding by finding the minimum distance between the input glyph codes and the models representing font-encodings. Existing works (Anil Kumar Singh and Jagadeesh Gorla, 2007) addressed the

Table 1: Font-Type Identification for Words.

Font Name	Uniglyph	Biglyph	Triglyph
Amarujala (Hindi)	100%	100%	100%
Jagran (Hindi)	100%	100%	100%
Webdunia (Hindi)	0.1%	100%	100%
Shree-Tel (Telugu)	7.3%	100%	100%
Eenadu (Telugu)	0.2%	100%	100%
Vaarthha (Telugu)	29.1%	100%	100%
E-Panchali (Tamil)	93%	100%	100%
Amudham (Tamil)	100%	100%	100%
Shree-Tam (Tamil)	3.7%	100%	100%
English-Text	0%	96.3%	100%

same problem but with limited success.

In this context, the proposed approach (A. A. Raj and K. Prahallad, 2007) use vector space model and Term Frequency - Inverse Document Frequency (TF-IDF) for font-encoding identification. This approach is used to weigh each term in the font-data according to its uniqueness. Thus it captures the relevancy among term and document. Here, *Term*: refers to a unit of glyph. In this work, experiments are performed with different units such as single glyph g_i (uniglyph), two consecutive glyphs $g_{i-1}g_i$ (biglyph) and three consecutive glyphs $g_{i-1}g_i g_{i+1}$ (triglyph). *Document*: It refers to the font-data (words and sentences) in a specific font-encoding.

To build a model for each font-encoding scheme, we need sufficient data. So we have collected manually an average of 0.12 million unique words per type for nearly 37 different glyph based fonts. To create a vector space model for a font-encoding, primarily the term (uniglyph or biglyph or triglyph) is extracted out of the font-data. Then TF-IDF weights are calculated for all terms in the documents.

Identification Results: The steps involved are as follows. Firstly, terms from the input word or sentence are extracted. Then a query vector using those terms is created. The distance between query vector and all the models of font-encoding is computed using TF-IDF weights. The input word is said to be originated from the model which gives a maximum TF-IDF value. It is typically observed that TF-IDF weights are more sensitive to the length of query. The accuracy increases with the increase in the length of test data. Thus, two different types

Table 2: Font-Type Identification for Sentences.

Font Name	Uniglyph	Biglyph	Triglyph
Amarujala (Hindi)	100%	100%	100%
Jagran (Hindi)	100%	100%	100%
Webdunia (Hindi)	100%	100%	100%
Shree-Tel (Telugu)	100%	100%	100%
Eenadu (Telugu)	0%	100%	100%
Vaarttha (Telugu)	100%	100%	100%
E-Panchali (Tamil)	100%	100%	100%
Amudham (Tamil)	100%	100%	100%
Shree-Tam (Tamil)	100%	100%	100%
English-Text	0%	100%	100%

of test data were prepared for testing. One is a set of unique words and the other is a set of sentences. It should also be noted that the accuracy depends on various factors: a) The number of font-encodings from which the identifier has to select one b) The inherent confusion of one font-encoding with another and c) The type of unit used in modeling. For 1000 different number of inputs (words and sentences) we have identified the closest models and calculated the accuracy. It is repeatedly done for various (uniglyph, biglyph and triglyph) categories. From Tables 1 and 2, it is clear that triglyph seems to be an appropriate unit for a term in the identification of font-encoding. It can also be seen that the performance at word and sentence level is 100% with triglyph.

4.2 Font-Data Conversion

The problem of font-data conversion could be defined as a module whose input is sequence of glyph codes and whose output is a sequence of Aksharas (characters) of Indian languages.

Existing methods and solutions proposed by (Himanshu Garg, 2005) (Khudanpur S. and Schafer C., Devanagari Converters, 2003) lack in, a) Framing a generic methodology or algorithm for conversion of font-data of all Indian languages b) Since glyph codes are manually constructed, 100% accurate conversion is achievable c) Existing methods requires large amount of effort for each font-encoding d) Large number of rules have to be written for rule based system e) Large parallel corpora has to be prepared for training f) They don't exploit shape and positional information of the glyphs, thus reducing

accuracy in conversion process.

Exploiting Position and Shape Information: (A. A. Raj and K. Prahallad, 2007) Characters in Indian languages change their shape where they appear (top, bottom, left, right) in the script. In this work, an unambiguous glyph code mapping is done by introducing a simple alphanumeric scheme where the alphabets denote the corresponding phoneme and the number denotes the glyph position. We followed IT3 phonetic notations and the position numbers as described below. Glyphs which could be in a) pivotal (center) position are referred by code 0/1. b) left position of pivotal symbol are referred by code 2. c) right position of pivotal symbol are referred by code 3. d) top position of pivotal symbol are referred by code 4. e) bottom position of pivotal symbol are referred by code 5.

Training: First in the training, a font-encoding for a language is selected and a glyph-map table is prepared by hand-coding the relation between glyphs (suffixed with their position and shape information) and IT3 notations. In the second stage, a simple set of glyph assimilation rules are defined (Multi-Lingual Screen Reader and Processing of Font-data in Indian Languages,). We iterated through the following steps until there are minimal errors on held-out test set of words. Results are checked for errors using human evaluation. If errors are found then the rules are updated or redefined. The above process is repeated for 3 different font-encodings of different font-families of the chosen language.

Evaluation: While testing, a new font from the same language is selected and a glyph-mapping table is prepared. It has to be noted that for new font, we don't update or add any glyph assimilation rules, and thus we use the existing rules obtained during training phase. A random set of 500 words from that font-data is picked-up. The conversion accuracy is evaluated using human evaluation. We have built converters for 10 Indian languages and 37 different font-encodings. The evaluations results in Table 3 indicate that the font-data conversion performs consistently above 99% for a new font-encoding across languages except for Telugu. Thus in our approach the effort of building rules is limited to three different fonts of a language to build the converter. To add a new font, only glyph-map table is required and no more repetition of rule building process.

In Table 3, we can observe inferior performance for Telugu. It is due to the number of glyphs and their possible combinations are higher than other languages. Also it is common for all Indian languages that the pivotal character glyph comes first and other supporting glyphs come next in the script. But in Telugu the supporting glyphs may come before the pivotal glyph which creates ambiguity in forming assimilation rules.

5 Experiments and Discussion

In this section, the experiments performed to build the tools/modules are explained. Most of them used the CART tool to train and test. These modules/tools are integrated and used for development of the proposed search engine.

5.1 CART (Classification and Regression Tree)

CART is a decision tree procedure introduced by Breiman et al., in 1984. CART uses an exhaustive, recursive partitioning routine to generate binary splits that divide each parent node into two child nodes by posing a series of yes-no questions. CART searches for questions that split nodes into relatively homogenous child nodes. As the tree evolves, the nodes become increasingly more homogenous, identifying segments. The basic CART building algorithm is a greedy algorithm which chooses the locally best discriminatory feature at each stage in the process.

Stop Parameter: The stop parameter specifies the minimum number of samples necessary in the training set before a question is hypothesized to distinguish the group. Normally with smaller stop value the model may become over-trained. The optional stop value may differ for different datasets of different languages.

Predictee: In a given feature set, the feature that is to be predicted as the output is termed as the predictee. By default, the first feature in the feature-set is taken as the predictee, but always the predictee can be specified while giving the description of the data. Some times CART is over-fit with training data, which may reduce the performance.

Feature Selection: Many experiments were conducted for different problems like grapheme to phoneme conversion (G2P) for English (Indian-

Table 3: Font-Data Conversion Results (Precision Values).

Language	Font Name	Training / Testing	Result
Hindi	Amarujala	Training	99.2%
	Jagran	Training	99.4%
	Naidunia	Training	99.8%
	Webdunia	Training	99.4%
	Chanakya	Testing	99.8%
Marathi	ShreePudhari	Training	100%
	ShreeDev	Training	99.8%
	TTYogesh	Training	99.6%
	Shusha	Testing	99.6%
Telugu	Eenadu	Training	93%
	Vaarttha	Training	92%
	Hemalatha	Training	93%
	TeluguFont	Testing	94%
Tamil	ElangoValluvan	Training	100%
	ShreeTam	Training	99.6%
	ElangoPanchali	Training	99.8%
	Tboomis	Testing	100%
Kannada	Shree Kan	Training	99.8%
	TTNandi	Training	99.4%
	BRH Kannada	Training	99.6%
	BRH Vijay	Testing	99.6%
Malayalam	Revathi	Training	100%
	Karthika	Training	99.4%
	Thoolika	Training	99.8%
	Shree Mal	Testing	99.6%
Gujarati	Krishna	Training	99.6%
	Krishnaweb	Training	99.4%
	Gopika	Training	99.2%
	Divaya	Testing	99.4%
Punjabi	DrChatrikWeb	Training	99.8%
	Satluj	Training	100%
Bengali	ShreeBan	Training	97.5%
	hPrPfPO1	Training	98%
	Aajkaal	Training	96.5%
Oriya	Dharitri	Training	95%
	Sambad	Training	97%
	AkrutiOri2	Training	96%

English) and stemming. These experiments were conducted with different possible features and stop values. Features for English G2P conversion were manually prepared for each letter and for stemming, the roots were manually identified for each word. The features vary from experiment to experiment and consequently the dimension of the features also vary.

Evaluation: For each experiment, we have considered 'N' number of words per language and we have generated 'M' number of features out of them. From the available features we have segregated 'X' number of features for training and 'Y' number of features for testing in 80:20 ratio. Using these sets, we have evaluated the training and testing performance for various stop values.

5.2 Stemming

Stemming is the use of linguistic analysis to get to the root form of a word. Search engines that use stemming compare the root forms of the search terms to the documents in its database. For example, if the user enters “viewer” as the query, the search engine reduces the word to its root (“view”) and returns all documents containing the root - like documents containing view, viewer, viewing, preview, review etc. Since our training data is very small it fails for out-of-vocabulary words. And also, it fails for homographs (a homograph is one of a group of words that share the same spelling but have different meanings).

For stemming in Indian languages, inflections of the words are formed mostly by suffixes than prefixes. So considering the first 5 phones of a word would help to predict the root of the word. But, for English prefixes as well as suffixes are equally used to form inflections. So prefixes are separated and considered as a single unit like a phone here. So we have selected the features like

- First 6 phones for English and
- First 5 phones for Indian languages

Stemming results for various languages are shown in Table 4. It shows that stop-value 1 would be optimal, when we used training and testing features in the ratio 907:227 for English, 3606:902 for Tamil and 987:247 for Telugu.

Table 4: Stemming Performance.

Language	Stop Value	Training	Testing
English	1	100%	99.55%
	2	98.78%	96.01%
	3	94.59%	90.26%
	4	86.64%	82.3%
Tamil	1	93.25%	77.69%
	2	84.74%	75.24%
	3	80.63%	74.49%
	4	77.08%	73.47%
Telugu	1	100%	93%
	2	100%	92%
	3	100%	93%
	4	100%	94%

5.3 English G2P Conversion

Our search uses phonetic features like syllables. In cross-linguagl search support for English input is necessary. So we need a mechanism to convert the query from its grapheme form to phoneme form. It is very challenging since English words doesn't follow one-to-one correspondence between its letters and its phonemes.

For G2P conversion of English words, the letters of the word are used as features. We hypothesize that the first and last letters of the word and previous and next letters of the current letter help much to predict its phoneme. So we have selected the features like

- First and Last letters of the word and Previous and Next letters of the Current letter

The G2P conversion results for Indian-English is shown in Table 5. It shows that stop-value 1 would be optimal for a training feature set of 106896 and testing feature set of 26724.

5.4 Stop Words Identification

Stop words, is the name given to the words which are filtered out prior to, or after processing of natural language data (text). There is no definite list of stop words which all natural language processing tools incorporate. Some search engines don't index/record extremely common words in order to save space or to speed up searches. The list of stop

Table 5: English G2P Conversion Performance.

Stop Value	Training	Testing
1	95.89%	85.56%
2	92.15%	85.37%
3	90.79%	85.56%
4	89.73%	85.53%

words for Indian languages have not been identified yet. So, we tried to generate the list by the basic idea that the most common words of a language might have occurred more frequently than other words in the corpus. We generated a list of top 500 frequently occurred words in a language. Then stop words list was produced with the help of a linguist who manually cleaned it.

6 inSearch - Search Engine for Indian Languages

Most information seekers use a search engine to begin their web activity (Prasad Pingali, Jagadeesh Jalagarlamudi and Vasudeva Varma, 2006). In this case, users submit a query (typically a list of keywords) and receive a list of web pages that may be relevant. In conventional information retrieval systems (Google, Live, Yahoo, Guruji etc.) the user must enter a search query in the language/encoding of the documents in order to retrieve it. This restriction clearly limits the amount of information to which an user will have access.

Developing a search engine for Indian languages faces many challenges. Some of them are, identifying the text-encoding of the web content, converting them into a transliteration scheme, developing stemmers and identifying the stop words etc. Also one need to design a good mechanism/tool (A. Joshi, A. Ganu, A. Chand, V. Parmar and G. Mathur, 2004) to accept user’s query in transliteration scheme or standard encoding like UTF-8 and even in English also. **inSearch** is a search engine for Indian languages developed by considering all the above discussed issues and solutions. Fig 1 shows the basic architecture and the following sub-sections explain them further.

6.0.1 Web Crawling

Our web crawling is language focused. It takes a list of identified URLs per language for which we have converters. Then it crawls those pages and stores the documents locally for further processing. It maintains the same directory structure as on the web and ordered by date.

6.0.2 Indexing

The effectiveness of any search engine mainly depends on its index structure. The structure should be capable of catering sufficient and relevant information to the users in terms of their needs. To serve users with the contextual information in their own language, the system needs to index on meaning representation and not on plain text. Also, the size of the index should not be too large.

Conventional search engines use stemming technology to get the root of the word and index the document about it. Thus, it will search not only for the search terms, but also for its inflexions and similar to some or all of those terms. But in case of Indian languages, there is no effective algorithm or tool to do stemming. So we used phonetic features like syllables to index the pages. We extract the first two syllables (since they are almost equal to the root of the word most of the times) of the word and index about it. Since, we have identified a method for stemming, we used them also for indexing. The detailed experiments are provided in the above section 5.2. Our index structure includes syllables, stem, word, term-frequency, language, date and doc-id. This structure enables efficient multi-lingual and cross-lingual search.

6.0.3 Retrieval

At first, beginning two syllables of the words of the query are extracted. Then the words beginning with those syllables are retrieved from the database. Hence the common words across languages are captured here. These words are ranked according to their phonetic relativeness to the query calculated by DTW method. The words fall under threshold are discarded, so that the documents containing the most related words pop-up. Then the documents are re-ranked about their term frequency (TF) values (G. Salton and C. Buckley, 1988) and contextual information.

6.0.4 User Interface

Presently, there is no standard/convenient notation or keyboard layout to input the query in Indian languages. Even with UTF-8 encoding most of the users don't know how to type the query. So, for cross-lingual search we provide a phonetic mapping table to be referred by the user to type the query in IT3 notation. But for language specific search, we provide a query typing tool. This tool has buttons for characters of the selected language. By clicking the buttons, user can type the query in his native script since most of the queries won't be more than a word/phrase. After forming the query, user can search the web and the ranked results are displayed much like the standard search engine's results. Here the cached pages for even font encoded pages are displayed in UTF-8 encoding.

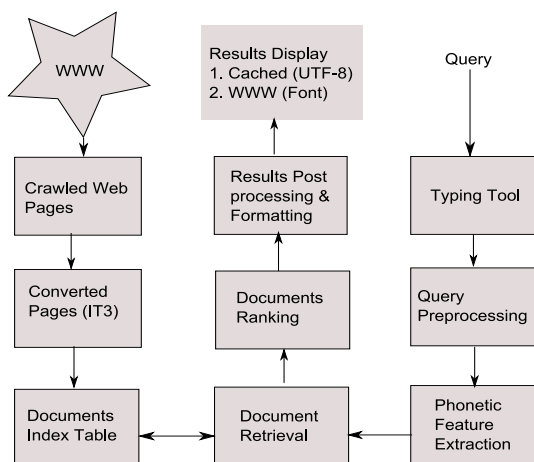


Figure 1: Search Engine Architecture.

7 Cross-Lingual Search

The development of digital and online information repositories has created many opportunities and new problems in information retrieval. Online documents are available Internationally in many different languages. This makes it possible for users to directly access previously unimagined sources of information. However in conventional information retrieval systems, the user must enter a search query in the language of the documents in order to retrieve it. This restriction clearly limits the amount and type of information which an individual user really has access to. Cross Language Information Retrieval

(CLIR) (F. Gey, N. Kando and C. Peters, 2002) (L. S. Larkey, M. S. Connell and N. Abduljaleel, 2003) enables users to enter queries in languages they are fluent in, and uses language translation methods to retrieve documents originally written in other languages.

The aim of this attempt is to extend the search capability to search across all Indian languages. The users are ordinary Indians who master one of the Indian languages, but have only passive knowledge in the other neighbour languages. This means that they can read a text but not search for it since they do not have active knowledge of how the different concepts in the other languages are written or spelled. This will also strengthen the use of the Indian languages on the Internet and further avoid unnecessary use of the English language. We are trying to achieve it step-by-step by using the below mentioned methods.

1. Phonetic Relativeness Measure: In this approach the phonetic distance (how many insertions/substitutions/deletions occurred) between the query words and the available words is calculated. Then the closest words are considered as query related words and the results are produced for those words. There are many methods to calculate the phonetic distance and we used DTW (Dynamic Time Warping) method to calculate the phonetic distance for our experiments. We used equal weightage (i.e 1) for insertion, substitution and deletion here.

2. Dictionary Lookup: Here bilingual/multilingual dictionaries are used to get the translation of the keywords. Creating such dictionaries for all the words of a language is time consuming process. Instead, creating dictionaries for the stems of the words alone will reduce the effort. Unfortunately there are no such dictionaries available or methods to create the stems for all Indian languages. So we developed CART based decision trees to produce the stems. We have created such stem based bilingual dictionaries for 5 Indian languages. Also, we have created a multilingual dictionary (Table 6) for 8 Indian languages by keeping English words as keys.

3. Machine Translation: This is considered as an appropriate solution for cross-language search (Dr. Pushpak Bhattacharyya, 2006). The query in source language gets translated into the destination language and the results will be produced for it. In this context, there is a close synergy between the fields of

Table 6: Multi-lingual Dictionary.

Language	Words
Bengali	2028
Gujarati	6141
Hindi	22212
Kannada	22695
Malayalam	23338
Oriya	7287
Tamil	5521
Telugu	8148
English	43185

Cross Language Information Retrieval (CLIR) and Machine Translation (MT). But such systems for Indian languages are under development. We are also focussing our effort in the same direction to use it with our engine in the future.

8 Conclusion

In this paper we discussed the importance of being able to search the Indian language web content and presented a multi-lingual web search engine in-Search capable of searching 10 Indian languages. The nature of Indic scripts, Indic data storage formats and how to preprocess them efficiently are detailed. It explained about how language identification, grapheme to phoneme conversion for English and stemming can be achieved using CART. This shows that transcoding of proprietary encodings into a meta standard transliteration scheme makes Indian language web content accessible through search engines.

9 Acknowledgments

We like to thank Speech and Language Technologies Lab, Bhriqus (India) Pvt Ltd, Hyderabad, India and our colleagues Ms.Bhuvaneshwari, Mr.Prasad and others for all their support and encouragement.

References

A. Joshi, A. Ganu, A. Chand, V. Parmar and G. Mathur. 2004. Keylekh: a keyboard for text entry in indic

scripts. *CHI '04 Extended Abstract on Human Factors in Computing Systems*, ACM Press.

- A. A. Raj and K. Prahallad. 2007. Identification and conversion of font-data in indian languages. In *In International Conference on Universal Digital Library (ICUDL)*, Pittsburgh, USA.
- A. K. Singh and J. Gorla. 2007. Identification of languages and encodings in a multilingual document. In *Proceedings of the 3rd ACL SIGWAC Workshop on Web As Corpus*, Louvain-la-Neuve, Belgium.
- Dr. P. Bhattacharyya. 2006. White paper on cross lingual search and machine translation. *Proposal to Government of India*.
- F. Gey, N. Kando and C. Peters. 2002. Cross language information retrieval: A research roadmap. *SIGIR Forum*, 36(2):72–80.
- G. E. Burkhart, S. E. Goodman, A. Mehta and L. Press. 1998. The internet in india: Better times ahead? *Commun. ACM*, 41(11):21–26.
- G. Salton and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Process. Management*, 24(5):513–523.
- M. Ganapathiraju , M. Balakrishnan , N. Balakrishnan and R. Reddy 2005. Om: One tool for many (indian) languages. *Journal of Zhejiang University Science*, 6A(11):1348–1353.
- H. Garg. 2005. Overcoming the font and script barriers among indian languages. *MS Thesis at International Institute of Information Technology Hyderabad, India*.
- ISCI - Indian Standard Code for Information Interchange. 1983. <http://tdil.mit.gov.in/standards.htm>.
- S.Khudanpur and C.Schafer , Devanagari Converters. 2003. http://www.cs.jhu.edu/cschafer/jhu_devanagari_cvt_ver2.tar.gz.
- L. S. Larkey, M. S. Connell and N. Abduljaleel. 2003. Hindi clir in thirty days. *ACM Trans. on Asian Language Information Processing (TALIP)*, 2(2):130–142.
- P. Lavanya, P. Kishore and G. R. Madhavi. 2005. A simple approach for building transliteration editors for indian languages. *Journal of Zhejiang University Science*, 6A(11):1354–1361.
- P. Prasad , J. Jagadeesh and V. Varma. 2006. Webkhaj: Indian language ir from multiple character encodings. *International World Wide Web Conference*.
- Transliteration of Indic Scripts: How to use ISO 15919. <http://homepage.ntlworld.com/stone-catend/trind.htm>.
- Unicode Consortium - Universal Code Standard. 1991. <http://www.unicode.org>.
- Multi-Lingual Screen Reader and Processing of Font-data in Indian Languages. <http://speech.iit.net/speech/publications/Anand-Thesis-Final.pdf>.

Cross-lingual Alignment and Completion of Wikipedia Templates

Gosse Bouma

Information Science
University of Groningen
g.bouma@rug.nl

Sergio Duarte

Information Science
University of Groningen
sergio.duarte@gmail.com

Zahurul Islam

Information Science
University of Groningen
zaisdb@gmail.com

Abstract

For many languages, the size of Wikipedia is an order of magnitude smaller than the English Wikipedia. We present a method for cross-lingual alignment of template and infobox attributes in Wikipedia. The alignment is used to add and complete templates and infoboxes in one language with information derived from Wikipedia in another language. We show that alignment between English and Dutch Wikipedia is accurate and that the result can be used to expand the number of template attribute-value pairs in Dutch Wikipedia by 50%. Furthermore, the alignment provides valuable information for normalization of template and attribute names and can be used to detect potential inconsistencies.

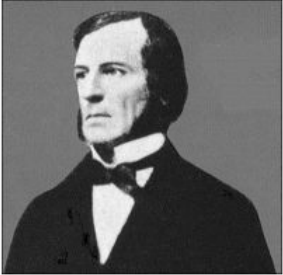
1 Introduction

One of the more interesting aspects of Wikipedia is that it has grown into a multilingual resource, with Wikipedia's for many languages, and systematic (cross-language) links between the information in different language versions. Eventhough English has the largest Wikipedia for any given language, the amount of information present in Wikipedia exceeds that of any single Wikipedia. One of the reasons for this is that each language version of Wikipedia has its own cultural and regional bias. It is likely, for instance, that information about the Netherlands is better represented in Dutch Wikipedia than in other Wikipedia's. Some indication that this is indeed the case comes from the fact a Google search for '*Pim Fortuyn*' in the Dutch Wikipedia gives 498 hits,

whereas the English Wikipedia gives only 292 hits. Also, 21,697 pages in Dutch Wikipedia fall in a category matching '*Nederlands(e)*', whereas only 9,494 pages in English Wikipedia fall in a category matching '*Dutch*'. This indicates that, apart from the obvious fact that smaller Wikipedia's can be expanded with information found in the larger Wikipedia's, it is also true that even the larger Wikipedia's can be supplemented with information harvested from smaller Wikipedia's.

Wikipedia infoboxes are tabular summaries of the most relevant facts contained in an article. They represent an important source of information for general users of the encyclopedia. Infoboxes (see figure 1) encode facts using attributes and values, and therefore are easy to collect and process automatically. For this reason, they are extremely valuable for systems that harvest information from Wikipedia automatically, such as DbPedia (Auer et al., 2008). However, as Wu and Weld (2007) note, infoboxes are missing for many pages, and not all infoboxes are complete. This is particularly true for Wikipedia's in languages other than English.

Infoboxes are a subclass of Wikipedia templates, which are used by authors of Wikipedia pages to express information in a systematic way, and to ensure that formatting of this information is consistent across Wikipedia. Templates exist for referring to multimedia content, external websites, news stories, scientific sources, other on-line repositories (such as the Internet Movie Database (IMDB), medical classification systems (ICD9 and ICD10), coordinates on Google Maps, etc. Although we are primarily interested in infoboxes, in the experiments below we take

Western Philosophy 19th-century philosophy	
	
George Boole	
Full name	George Lawlor Boole
Birth	November 2, 1815 (Lincoln Lincolnshire , England)
Death	December 8, 1864 (aged 49) (Ballintemple, County Cork, Ireland),(Drug Overdose)
School/tradition	Mathematical foundations of computer science
Main interests	Mathematics, Logic, Philosophy of mathematics
Notable ideas	Boolean algebra

```

{{Infobox Philosopher |
region = Western Philosophy |
era = [[19th-century philosophy]] |
image_name = George Boole.jpg|
image_caption = George Boole |
name = George Lawlor Boole |
birth = November 2, 1815 |
death = December 8, 1864 |
school = Mathematical foundations
of [[computer science]] |
main_interests = [[Mathematics]], [[Logic]] |
ideas = [[Boolean algebra]]
}}

```

Figure 1: Infobox and (simplified) Wikimedia source

all templates into account.

We plan to use information mined from Wikipedia for Question Answering and related tasks. In 2007 and 2008, the CLEF question answering track¹ used Wikipedia as text collection. While the usual approach to open domain question answering relies on information retrieval for selecting relevant text snippets, and natural language processing techniques for answer extraction, an alternative stream of research has focussed on the potential of on-line data-sets for question answering (Lita et al., 2004; Katz et al., 2005). In Bouma et al. (2008) it is suggested that information harvested from infoboxes can be used for question answering in CLEF. For instance, the answer to questions such as *How high is the Matterhorn?*, *Where was Piet Mondriaan born?*, and *What is the area of the country Suriname?* can in principle be found in infoboxes. However, in practice the number of questions that is answered by their Dutch QA-system by means information from infoboxes is small. One reason for this is the lack of coverage of infoboxes in Dutch Wikipedia.

¹<http://clef-qa.itc.it>

In the recent GIKICLEF task² systems have to find Wikipedia pages in a number of languages which match descriptions such as *Which Australian mountains are higher than 2000 m?*, *French bridges which were in construction between 1980 and 1990*, and *African capitals with a population of two million inhabitants or more*. The emphasis in this task is less on answer extraction from text (as in QA) and more on accurate interpretation of (geographical) facts known about an entity. GIKICLEF is closely related to the *entity ranking* task for Wikipedia, as organized by INEX.³ We believe systems participating in tasks like this could profit from large collections of $\langle \text{entity}, \text{attribute}, \text{value} \rangle$ triples harvested from Wikipedia templates.

In this paper, we propose a method for automatically expanding the amount of information present in the form of templates. In our experiments, we used English and Dutch Wikipedia as sources. Given a page in English, and a matching page in Dutch, we first find all English-Dutch attribute-value

²<http://www.linguateca.pt/GikiCLEF>

³<http://inex.is.informatik.uni-duisburg.de>

tuples which have a matching value. Based on the frequency with which attributes match, we create a bidirectional, intersective, alignment of English-Dutch attribute pairs. Finally, we use the set of aligned attributes to expand the number of attribute-value pairs in Dutch Wikipedia with information obtained from matching English pages. We also show that aligned attributes can be used to normalize attribute names and to detect formatting issues and potential inconsistencies in attribute values.

2 Previous Work

DbPedia (Auer et al., 2008) is a large, on-going, project which concentrates on harvesting information from Wikipedia automatically, on normalization of the extracted information, on linking the information with other on-line data repositories, and on interactive access. It contains 274M facts about 2.6M entities (November, 2008). An important component of DbPedia is harvesting of the information present in infoboxes. However, as Wu and Weld (2007) note, not all relevant pages have (complete) infoboxes. The information present in infoboxes is typically also present in the running text of a page. One line of research has concentrated on using the information obtained from infoboxes as seeds for systems that learn relation extraction patterns (Nguyen et al., 2007). Wu and Weld (2007) go one step further, and concentrate on learning to complete the infoboxes themselves. They present a system which first learns to predict the appropriate infobox for a page (using text classification). Next, they learn relation extraction patterns using the information obtained from existing infoboxes as seeds. Finally, the learned patterns are applied to text of pages for which a new infobox has been predicted, to assign values to infobox attributes. A recent paper by Adar et al. (2009) (which only came to our attention at the time of writing) starts from the same observation as we do. It presents a system for completing infoboxes for English, German, French, and Spanish Wikipedia, which is based on learning a mapping between infoboxes and attributes in multiple languages. A more detailed comparison between their approach and ours is given in section 6

The potential of the multilingual nature of Wikipedia has been explored previously by several

researchers. Adafre and de Rijke (2006) explore machine translation and (cross-lingual) link structure to find sentences in English and Dutch Wikipedia which express the same content. Bouma et al. (2006) discuss a system for the English-Dutch QA task of CLEF. They basically use a Dutch QA-system, which takes questions automatically translated from English (by the on-line Babelfish translation service). To improve the quality of the translation of named entities, they use, among others, cross-language links obtained from Wikipedia. Erdmann et al. (2008) explore the potential of Wikipedia for the extraction of bilingual terminology. They note that apart from the cross-language links, page redirects and anchor texts (i.e. the text that is used to label a hypertext reference to another (wikipedia) page) can be used to obtain large and accurate bilingual term lists.

3 Data collection and Preparation

We used a dump of Dutch Wikipedia (June 2008) and English Wikipedia (August 2007) made available by the University of Amsterdam⁴ and converted to an XML-format particularly suitable for information extraction tasks.

From these two collections, for each page, we extracted all attribute-value pairs found in all templates. Results were stored as quadruples of the form $\langle Page, TemplateName, Attribute, Value \rangle$. Each $TemplateName \sim Attribute$ pair expresses a specific semantic relation between the entity or concept described by the *Page* and a *Value*. Values can be anything, but often refer to another Wikipedia page (i.e. $\langle George\ Boole, Philosopher, notable_ideas, Boolean\ algebra \rangle$, where *Boolean algebra* is a link to another page) or to numeric values, amounts, and dates. Note that attributes by themselves tend to be highly ambiguous, and often can only be interpreted in the context of a given template. The attribute *period*, for instance, is used to describe chemical elements, royal dynasties, countries, and (historical) means of transportation. Another source of attribute ambiguity is the fact that many templates simply number their attributes. As we are interested in finding an alignment between semantically meaningful relations in the two collections, we will therefore

⁴<http://ilps.science.uva.nl/WikiXML>

	Dutch	English
date	June 2008	August 2007
pages	715,992	3,840,950
pages with template	290,964	757,379
cross-language links		126,555
templates	550,548	1,074,935
tuples	4,357,653	5,436,033
template names	2,350	7,783
attribute names	7,510	19,378
templ~attr pairs	23,399	81,671

Table 1: Statistics for the version of Dutch and English Wikipedia used in the experiment.

concentrate on the problem of finding an alignment between *TemplateName~Attribute* pairs in English and Dutch Wikipedia.

Some statistics for the two collections are given in table 1. The number of pages is the count for all pages in the collection. It should be noted that these contain a fair number of administrative pages, pages for multimedia content, redirect pages, page stubs, etc. The number of pages which contains content that is useful for our purposes is therefore probably a good deal lower, and is maybe closer to the number of pages containing at least one template. Cross-language links (i.e. links from an English page to the corresponding Dutch page) were extracted from English Wikipedia. The fact that 0.5M templates in Dutch give rise to 4.3M tuples, whereas 1.0M templates in English give rise to only 5.4M tuples is perhaps a consequence of the fact that the two collections are not from the same date, and thus may reflect different stages of the development of the template system.

We did spend some time on normalization of the values found in extracted tuples. Our alignment method relies on the fact that for a sufficient number of matching pages, tuples can be found with matching values. Apart from identity and Wikipedia cross-language links, we rely on the fact that dates, amounts, and numerical values can often be recognized and normalized easily, thus increasing the number of tuples which can be used for alignment. Normalization addresses the fact that the use of comma’s and periods (and spaces) in numbers is different in English and Dutch Wikipedia, and that

dates need to be converted to a standard. English Wikipedia expresses distances and heights in miles and feet, weights in pounds, etc., whereas Dutch Wikipedia uses kilometres, metres, and kilograms. Where English Wikipedia mentions both miles and kilometres we preserve only the kilometres. In other situations we convert miles to kilometres. In spite of this effort, we noted that there are still quite a few situations which are not covered by our normalization patterns. Sometimes numbers are followed or preceded by additional text (*approx.14.5 MB, 44 minutes per episode*), sometimes there is irregular formatting (*October 101988*), and some units simply are not handled by our normalization yet (i.e. converting square miles to square kilometres). We come back to this issue in section 6.

Kröttsch et al. (2007) have also observed that there is little structure in the way numeric values, units, dates, etc. are represented in Wikipedia. They suggest a tagging system similar to the way links to other Wikipedia pages are annotated, but now with the aim of representing numeric and temporal values systematically. If such a system was to be adopted by the Wikipedia community, it would greatly facilitate the processing of such values found in infoboxes.

4 Alignment

In this section we present our method for aligning English *TemplateName~Attribute* pairs with corresponding pairs in Dutch.

The first step is creating a list of matching tuples.

Step 1. Extract all matching template tuples.

An English $\langle Page_e, Templ_e \sim Attr_e, Val_e \rangle$ tuple matches a Dutch $\langle Page_d, Templ_d \sim Attr_d, Val_d \rangle$ tuple if $Page_e$ matches $Page_d$ and Val_e matches Val_d and there is no other tuple for either $Page_e$ or $Page_d$ with value Val_e or Val_d .

Two pages or values E and D match if there exists a cross-language link which links E and D , or if $E=D$.

We only take into account tuples for which there is a unique (non-ambiguous) match between English and Dutch. Many infoboxes contain attributes which

often take the same value (i.e. *title* and *imdb_title* for movies). Other cases of ambiguity are caused by numerical values which incidentally may take on identical values. Such ambiguous cases are ignored. Step 1 gives rise to 149,825 matching tuples.⁵ It might seem that we find matching tuples for only about 3-4% of the tuples present in Dutch Wikipedia. Note, however, that while there are 290K pages with a template in Dutch Wikipedia, there are only 126K cross-language links. The total number of tuples on Dutch pages for which a cross-language link to English exists is 837K. If all these tuples have a counterpart in English Wikipedia (which is highly unlikely), our method finds a match for 18% of the relevant template tuples.⁶

The second step consists of extracting matching English-Dutch Template~Attribute pairs from the list of matching tuples constructed in step 1.

Step 2. For each matching pair of tuples $\langle Page_e, Templ_e \sim Attr_e, Val_e \rangle$ and $\langle Page_d, Templ_d \sim Attr_d, Val_d \rangle$, extract the English-Dutch pair of Template~Attributes $\langle Templ_e \sim Attr_e, Templ_d \sim Attr_d \rangle$.

In total, we extracted 7,772 different English-Dutch $\langle Templ_e \sim Attr_e, Templ_d \sim Attr_d \rangle$ tuples. In 547 cases $Templ_e \sim Attr_e = Templ_d \sim Attr_d$. In 915 cases, $Attr_e = Attr_d$. In the remaining 6,310 cases, $Attr_e \neq Attr_d$. The matches are mostly accurate. We evaluated 5% of the matching template~attribute pairs, that had been found at least 2 times. For 27% of these (55 out of 205), it was not immediately clear whether the match was correct, because one of the attributes was a number. Among the remaining 150 cases, the only clear error seemed to be a match between the attributes *trainer* and *manager* (for soccer club templates). Other cases which are perhaps not always correct were mappings between *successor*, *successor1*, *successor2* on the one hand and *after/next* on the other hand. The attributes with a

⁵It is interesting to note that 51K of these matching tuples are for pages that have an identical name in English and Dutch, but were absent in the table of cross-language links. As a result, we find 32K pages with an identical name in English and Dutch, and at least one pair of matching tuples. We suspect that these newly discovered cross-language links are highly accurate.

⁶If we also include English pages with a name identical to a Dutch page, the maximum number of matching tuples is 1.1M, and we find a match for 14% of the data

101	cite_web	title
27	voetnoot_web	titel
12	film	titel
10	commons	1
7	acteur	naam
6	game	naam
5	ster	naam
4	taxobox_zoogdier	w-naam
4	plaats	naam
4	band	band_naam
3	taxobox	w-naam

Table 2: Dutch template~attribute pairs matching English *cite_web~title*. Counts refer to the number of pages with a matching value.

number suffix probably refer to the *n*th successor, whereas the attributes without suffix probably refer to the immediate successor.

On the other hand, for some frequent template~attribute pairs, ambiguity is clearly an issue. For the English pair *cite_web~title* for instance, 51 different mappings are found. The most frequent cases are shown in table 2. Note that it would be incorrect to conclude from this that, for every English page which contains a *cite_web~title* pair, the corresponding Dutch page should include, for instance, a *taxobox~w-naam* tuple.

In the third and final step, the actual alignment between English-Dutch template~attribute pairs is established, and ambiguity is eliminated.

Step 3. Given the list of matching template~attribute pairs computed in step 2 with a frequency ≥ 5 , find for each English $Templ_e \sim Attr_e$ pair the most frequent matching Dutch pair $Templ_d \sim Attr_d$. Similarly, for each Dutch pair $Templ_d \sim Attr_d$, find the most frequent English pair $Templ_e \sim Attr_e$. Return the intersection of both lists.

2,070 matching template~attribute tuples are seen at least 5 times. Preference for the most frequent bidirectional match leaves 1,305 template~attribute tuples. Examples of aligned tuples are given in table 3. We evaluated 10% of the tuples containing meaningful attributes (i.e. not

English		Dutch	
Template	Attribute	Template	Attribute
actor	spouse	acteur	partner
book	series	boek	reeks
casino	owner	casino	eigenaar
csi_character	portrayed	csi_personage	acteur
dogbreed	country	hond	land
football_club	ground	voetbal_club	stadion
film	writer	film	schrijver
mountain	range	berg	gebergte
radio_station	airdate	radiozender	lancering

Table 3: Aligned template~attribute pairs

numbers or single letters). In 117 tuples, we discovered two errors: $\langle \text{aircraft_specification} \sim \text{number of props}, \text{gevechtsvliegtuig} \sim \text{bemanning} \rangle$ aligns the number of engines with the number of crew members (based on 10 matching tuples), and $\langle \text{book} \sim \text{country}, \text{film} \sim \text{land} \rangle$ involves a mismatch of templates as it links the country attribute for a book to the country attribute for a movie.

Note that step 3 is similar to bidirectional inter-sective word alignment as used in statistical machine translation (see Ma et al. (2008), for instance). This method is known for giving highly precise results.

5 Expansion

We can use the output of step 3 of the alignment method to check for each English tuple whether a corresponding Dutch tuple can be predicted. If the tuple does not exist yet, we add it. In total, this gives rise to 2.2M new tuples for 382K pages for Dutch Wikipedia (see table 4). We generate almost 300K new tuples for existing Dutch pages (250K for pages for which a cross-language link already existed). This means we expand the total number of tuples for existing pages by 27%. Most tuples, however, are generated for pages which do not yet exist in Dutch Wikipedia. These are perhaps less useful, although one could use the results as knowledge for a QA-system, or to generate stubs for new Wikipedia pages which already contain an infobox and other relevant templates.

The 100 most frequently added template~attribute pairs (ranging from $\text{music album} \sim \text{genre}$ (added 31,392 times) to $\text{single} \sim \text{producer}$ (added 5605

	pages	triples
existing pages	50,099	253,829
new cross-links	11,526	43,449
new dutch pages	321,069	1,931,277
total	382,694	2,228,555

Table 4: Newly inferred template tuples

times)) are dominated by templates for music albums, geographical places, actors, movies, and taxonomy infoboxes.

We evaluated the accuracy of the newly generated tuples for 100 random existing Dutch wikipedia pages, to which at least one new tuple was added. The pages contained 802 existing tuples. 876 tuples were added by our automatic expansion method. Of these newly added tuples, 62 contained a value which was identical to the value of an already existing tuple (i.e. we add the tuple $\langle \text{Reuzenhaai}, \text{taxobox} \sim \text{naam}, \text{Reuzenhaai} \rangle$ where there was already an existing tuple $\langle \text{Reuzenhaai}, \text{taxobox} \sim \text{begin} \sim \text{name}, \text{Reuzenhaai} \rangle$ tuple – note that we add a properly translated attribute name, where the original tuple contains a name copied from English!). The newly added tuples contained 60 tuples of the form $\langle \text{Aegna}, \text{plaats} \sim \text{lat_dir}, N(\text{letter}) \rangle$, where the value should have been N (the symbol for latitude on the Northern hemisphere in geographical coordinates), and not the letter N. One page (*Akira*) was expanded with an incoherent set of tuples, based on tuples for the manga, anime, and music producer with the same name. Apart from this failure, there were only 5 other clearly incorrect tuples (adding o.a. $\text{place} \sim \text{name}$ to *Albinism*, adding *name in Dutch* with an English value to *Macedonian*, and adding $\text{community} \sim \text{name}$ to *Christopher Columbus*). In many cases, added tuples are based on a different template for the same entity, often leading to almost identical values (i.e. adding geographical coordinates using slightly different notation). In one case, *Battle of Dogger Bank (1915)*, the system added new tuples based on a template that was already in use for the Dutch page as well, thus automatically updating and expanding an existing template.

geboren		population	
23	birth_date	49	inwoners
16	date of birth	9	population
8	date_of_birth	5	bevolking
8	dateofbirth	4	inwonersaantal
2	born	3	inwoneraantal
2	birth	2	town pop
1	date_birth	2	population_total
1	birthdate	1	townpop
		1	inw.
		1	einwohner

Table 6: One-to-many aligned attribute names. Counts are for the number of (aligned) infoboxes that contain the attribute.

6 Discussion

6.1 Detecting Irregularities

Instead of adding new information, one may also search for attribute-value pairs in two Wikipedia’s that are expected to have the same value, but do not. Given an English page with attribute-value pair $\langle Attr_e, Val_e \rangle$, and a matching Dutch page with $\langle Attr_d, Val_d \rangle$, where $Attr_e$ and $Attr_d$ have been aligned, one expects Val_e and Val_d to match as well. If this is not the case, something irregular is observed. We have applied the above rule to our dataset, and detected 79K irregularities. An overview of the various types of irregularities is given in table 5. Most of the non-matching values are the result of formatting issues, lack of translations, one value being more specific than the other, and finally, inconsistencies. Note that inconsistencies may also mean that one value is more recent than the other (population, (stadium) capacity, latest release data, spouse, etc.). A number of formatting issues (of numbers, dates, periods, amounts, etc.) can be fixed easily, using the current list of irregularities as starting point.

6.2 Normalizing Templates

It is interesting to note that alignment can also be used to normalize template attribute names. Table 6 illustrates this for the Dutch attribute *geboren* and the English attribute *population*. Both are aligned with a range of attribute names in the other language.

Such information is extremely valuable for applications that attempt to harvest knowledge from Wikipedia, and merge the result in an ontology, or attempt to use the harvested information in an application. For instance, a QA-system that has to answer questions about birth dates or populations, has to know which attributes are used to express this information. Alternatively, one can also use this information to normalize attribute-names. In that case, all attributes which express the birth date property could be replaced by *birth_date* (the most frequent attribute currently in use for this relation).

This type of normalization can greatly reduce the *noisy* character of the current infoboxes. For instance, there are many infoboxes in use for geographic locations, people, works of art, etc. These infoboxes often contain information about the same properties, but, as illustrated above, there is no guarantee that these are always expressed by the same attribute.

6.3 Alignment by means of translation

Template and attribute names in Dutch often are straightforward translations of the English name, e.g. *luchtvaartmaatschappij/airline*, *voetbalclub/football club*, *hoofdstad/capital*, *naam/name*, *postcode/postalcode*, *netnummer/area_code* and *opgericht/founded*. One might use this information as an alternative for determining whether two template~attribute pairs express the same relation.

We performed a small experiment on infoboxes expressing geographical information, using Wikipedia cross-language links and an on-line dictionary as multilingual dictionaries. We found that 10 to 15% of the attribute names (depending on the exact subset of infoboxes taken into consideration) could be connected using dictionaries. When combined with the attributes found by means of alignment, coverage went up to maximally 38%.

6.4 Comparison

It is hard to compare our results with those of Adar et al. (2009). Their method uses a Boolean classifier which is trained using a range of features to determine whether two values are likely to be equivalent (including identity, string overlap, link relations, translation features, and correlation of numeric values). Training data is collected automatically by

English	Attributes		Values		Type
	English	Dutch	English	Dutch	
capacity	capaciteit	23,400	23 400	formatting	
nm	lat_min	04	4	formatting	
date	date	1775-1783	1775–1783	formatting	
name	naam	African baobab	Afrikaanse baobab	translation	
artist	artiest	Various Artists	Verschillende artiesten	translation	
regnum	rijk	Plantae	Plantae (Planten)	specificity	
city	naam,	Comune di Adrara San Martino	Adrara San Martino	specificity	
birth_date	geboren	1934	1934-8-25	specificity	
imagepath_coa	wapenafbeelding	coa_missing.jpg	Alvaneu wappen.svg	specificity	
population	inwonersaantal	5345	5369	inconsistent	
capacity	capaciteit	13,152	14 400	inconsistent	
dateofbirth	geboortedatum	2 February 1978	1978-1-2	inconsistent	
elevation	hoogte	300	228	inconsistent	

Table 5: Irregular values in aligned attributes on matching pages

selecting highly similar tuples (i.e. with identical template and attribute names) as positive data, and a random tuple from the same page as negative data. The accuracy of the classifier is 90.7%. Next, for each potential pairing of template~attribute pairs from two languages, random tuples are presented to the classifier. If the ratio of positively classified tuples exceeds a certain threshold, the two template~attribute pairs are assumed to express the same relation. The accuracy of result varies, with matchings of template~attribute pairs that are based on the most frequent tuple matches having an accuracy score of 60%. They also evaluate their system by determining how well the system is able to predict known tuples. Here, recall is 40% and precision is 54%. The recall figure could be compared to the 18% tuples (for pages related by means of a cross-language link) for which we find a match. If we use only properly aligned template~attribute pairs, however, coverage will certainly go down somewhat. Precision could be compared to our observation that we find 149K matching tuples, and, after alignment, predict an equivalence for 79K tuples which in the data collecting do not have a matching value. Thus, for 228K tuples we predict an equivalent values, whereas this is only the case for 149K tuples. We would not like to conclude from this, however, that the precision of our method is 65%, as we observed in section 5 that most of the conflicting values are not inconsistencies, but more often the consequence of formatting irregularities, transla-

tions, variation in specificity, etc. It is clear that the system of Adar et al. (2009) has a higher recall than ours. This appears to be mainly due to the fact that their feature based approach to determining matching values considers much more data to be equivalent than our approach which normalizes values and then requires identity or a matching cross-language link. In future work, we would like to explore more rigorous normalization (taking the data discussed in section 5 as starting point) and inclusion of features to determine approximate matching to increase recall.

7 Conclusions

We have presented a method for automatically completing Wikipedia templates which relies on the multilingual nature of Wikipedia and on the fact that systematic links exist between pages in various languages. We have shown that matching template tuples can be found automatically, and that an accurate set of matching template~attribute pairs can be derived from this by using intersective bidirectional alignment. The method extends the number of tuples by 51% (27% for existing Dutch pages).

In future work, we hope to include more languages, investigate the value of (automatic) translation for template and attribute alignment, investigate alternative alignment methods (using more features and other weighting scheme’s), and incorporate the expanded data set in our QA-system for Dutch.

References

- S.F. Adafre and M. de Rijke. 2006. Finding similar sentences across multiple languages in wikipedia. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 62–69.
- E. Adar, M. Skinner, and D.S. Weld. 2009. Information arbitrage across multi-lingual Wikipedia. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 94–103. ACM New York, NY, USA.
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2008. Dbpedia: A nucleus for a web of open data. *The Semantic Web*, pages 722–735.
- Gosse Bouma, Ismail Fahmi, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedemann. 2006. The University of Groningen at QA@CLEF 2006: Using syntactic knowledge for QA. In *Working Notes for the CLEF 2006 Workshop*, Alicante.
- Gosse Bouma, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedemann. 2008. Question answering with Joost at QA@CLEF 2008. In *Working Notes for the CLEF 2008 Workshop*, Aarhus.
- M. Erdmann, K. Nakayama, T. Hara, and S. Nishio. 2008. An approach for extracting bilingual terminology from wikipedia. *Lecture Notes in Computer Science*, 4947:380.
- B. Katz, G. Marton, G. Borchardt, A. Brownell, S. Felshin, D. Loreto, J. Louis-Rosenberg, B. Lu, F. Mora, S. Stiller, et al. 2005. External knowledge sources for question answering. In *Proceedings of the 14th Annual Text REtrieval Conference (TREC'2005), November*.
- M. Krötzsch, D. Vrandečić, M. Völkel, H. Haller, and R. Studer. 2007. Semantic wikipedia. *Web Semantics: Science, Services and Agents on the World Wide Web*.
- L.V. Lita, W.A. Hunt, and E. Nyberg. 2004. Resource analysis for question answering. In *Association for Computational Linguistics Conference (ACL)*.
- Y. Ma, S. Ozdowska, Y. Sun, and A. Way. 2008. Improving word alignment using syntactic dependencies. In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 69–77.
- D.P.T. Nguyen, Y. Matsuo, and M. Ishizuka. 2007. Relation extraction from wikipedia using subtree mining. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, page 1414. AAAI Press.
- Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 41–50, New York, NY, USA. ACM.

Directions for Exploiting Asymmetries in Multilingual Wikipedia

Elena Filatova

Computer and Information
Sciences Department
Fordham University
Bronx, NY 10458, USA
filatova@cis.fordham.edu

Abstract

Multilingual Wikipedia has been used extensively for a variety of Natural Language Processing (NLP) tasks. Many Wikipedia entries (people, locations, events, etc.) have descriptions in several languages. These descriptions, however, are not identical. On the contrary, descriptions in different languages created for the same Wikipedia entry can vary greatly in terms of description length and information choice. Keeping these peculiarities in mind is necessary while using multilingual Wikipedia as a corpus for training and testing NLP applications. In this paper we present preliminary results on quantifying Wikipedia multilinguality. Our results support the observation about the substantial variation in descriptions of Wikipedia entries created in different languages. However, we believe that asymmetries in multilingual Wikipedia do not make Wikipedia an undesirable corpus for NLP applications training. On the contrary, we outline research directions that can utilize multilingual Wikipedia asymmetries to bridge the communication gaps in multilingual societies.

1 Introduction

Multilingual parallel corpora such as translations of fiction, European parliament proceedings, Canadian parliament proceedings, the Dutch parallel corpus are being used for training machine translation and paraphrase extraction systems. All of these corpora are parallel corpora.

Parallel corpora contain the same information translated from one language (the source language

of the text) into a set of pre-specified languages with the goal of preserving the information covered in the source language document. Translators working with fiction also carefully preserve the stylistic details of the original text.

Parallel corpora are a valuable resource for training NLP tools. However, they exist only for a small number of language pairs and usually in a specific context (e.g., legal documents, parliamentary notes). Recently NLP community expressed a lot of interest in studying other types of multilingual corpora.

The largest multilingual corpus known at the moment is World Wide Web (WWW). One part of particular interest is the on-line encyclopedia-style site, Wikipedia.¹ Most Wikipedia entries (people, locations, events, etc.) have descriptions in different languages. However, Wikipedia is not a parallel corpus as these descriptions are not translations of a Wikipedia article from one language into another. Rather, Wikipedia articles in different languages are independently created by different users.

Wikipedia does not have any filtering on who can write and edit Wikipedia articles. In contrast to professional encyclopedias (like *Encyclopedia Britannica*), Wikipedia authors and editors are not necessarily experts in the field for which they create and edit Wikipedia articles. The trustworthiness of Wikipedia is questioned by many people (Keen, 2007).

The multilinguality of Wikipedia makes this situation even more convoluted as the sets of Wikipedia contributors for different languages are not the same.

¹<http://www.wikipedia.org/>

Moreover, these sets might not even intersect. It is unclear how similar or different descriptions of a particular Wikipedia entry in different languages are. Knowing that there are differences in descriptions for the same entry and the ability to identify these differences is essential for successful communication in multilingual societies.

In this paper we present a preliminary study of the asymmetries in a subset of multilingual Wikipedia. We analyze the number of languages in which the Wikipedia entry descriptions are created; and the length variation for the same entry descriptions created in different languages. We believe that this information can be helpful for understanding asymmetries in multilingual Wikipedia. These asymmetries, in turn, can be used by NLP researchers for training summarization systems, and contradiction detection systems.

The rest of the paper is structured as follows. In Section 2 we describe related work, including the work on utilizing parallel corpora. In Section 3 we provide examples of our analysis for several Wikipedia entries. In Section 4 we describe our corpus, and the systematic analysis performed on this corpus. In Section 5 we draw conclusions based on the collected statistics and outline avenues for our future research.

2 Related Work

There exist several types of multilingual corpora (e.g., parallel, comparable) that are used in the NLP community. These corpora vary in their nature according to the tasks for which these corpora were created.

Corpora developed for multilingual and cross-lingual question-answering (QA), information retrieval (IR), and information extraction (IE) tasks are typically compilations of documents on related subjects written in different languages. Documents in such corpora rarely have counterparts in all the languages presented in the corpus (CLEF, 2000; Magnini et al., 2003).

Parallel multilingual corpora such as Canadian parliament proceedings (Germann, 2001), European parliament proceedings (Koehn, 2005), the Dutch parallel corpus (Macken et al., 2007), JRC-ACQUIS Multilingual Parallel Corpus (Steinberger et al.,

2006), and so on contain documents that are exact translations of the source documents.

Understanding the corpus nature allows systems to utilize different aspects of multilingual corpora. For example, Barzilay *et al.* (2001) use several translations of the French text of *Gustave Flaubert's* novel *Madame Bovary* into English to mine a corpus of English paraphrases. Thus, they utilize the creativity and language expertise of professional translators who used different wordings to convey not only the meaning but also the stylistic peculiarities of *Flaubert's* French text into English.

Parallel corpora are a valuable resource for training NLP tools. However, they exist only for a small number of language pairs and usually in a specific context (e.g., legal documents, parliamentary notes). Recently NLP community expressed a lot of interest in studying comparable corpora. Workshops on building and using comparable corpora have become a part of NLP conferences (LREC, 2008; ACL, 2009). A comparable corpus is defined as a set of documents in one to many languages, that are comparable in content and form in various degrees and dimensions.

Wikipedia entries can have descriptions in several languages independently created for each language. Thus, Wikipedia can be considered a comparable corpus.

Wikipedia is used in QA for answer extraction and verification (Ahn et al., 2005; Buscaldi and Rosso, 2006; Ko et al., 2007). In summarization, Wikipedia articles structure is used to learn the features for summary generation (Baidys et al., 2008).

Several NLP systems utilize the Wikipedia multilinguality property. Adafre *et al.* (2006) analyze the possibility of constructing an English-Dutch parallel corpus by suggesting two ways of looking for similar sentences in Wikipedia pages (using matching translations and hyperlinks). Richman *et al.* (2008) utilize multilingual characteristics of Wikipedia to annotate a large corpus of text with Named Entity tags. Multilingual Wikipedia has been used to facilitate cross-language IR (Schönhofen et al., 2007) and to perform cross-lingual QA (Ferrández et al., 2007).

One of the first attempts to analyze similarities and differences in multilingual Wikipedia is described in Adar *et al.* (2009) where the main goal

is to use self-supervised learning to align or/and create new Wikipedia infoboxes across four languages (English, Spanish, French, German). Wikipedia infoboxes contain a small number of facts about Wikipedia entries in a semi-structured format.

3 Analysis of Multilingual Wikipedia Entry Examples

Wikipedia is a resource generated by collaborative effort of those who are willing to contribute their expertise and ideas about a wide variety of subjects. Wikipedia entries can have descriptions in one or several languages. Currently, Wikipedia has articles in more than 200 languages. Table 1 presents information about the languages that have the most articles in Wikipedia: the number of languages, the language name, and the Internet Engineering Task Force (IETF) standard language tag.²

English is the language having the most number of Wikipedia descriptions, however, this does not mean that all the Wikipedia entries have descriptions in English. For example, entries about people, locations, events, etc. famous or/and important only within a community speaking in a particular language are not likely to have articles in many languages. Below, we list a few examples that illustrate this point. Of course, more work is required to quantify the frequency of such entries.

- the Wikipedia entry about Mexican singer and actress *Rocío Banquells* has only one description: in Spanish;
- the Wikipedia entry about a mountain ski resort *Falakro* in northern Greece has descriptions in four languages: Bulgarian, English, Greek, Nynorsk (one of the two official Norwegian standard languages);
- the Wikipedia entry about *Prioksko-Terrasny Nature Biosphere Reserve*, a Russia's smallest nature reserve, has descriptions in two languages: Russian and English;

Number of Articles	Language	IETF Tag
2,750,000+	English	en
750,000+	German French	de fr
500,000+	Japanese Polish Italian Dutch	jp pl it nl

Table 1: Language editions of Wikipedia by number of articles.

- the Wikipedia entry about a Kazakhstani figure skater *Denis Ten* who is of partial Korean descent has descriptions in four languages: English, Japanese, Korean, and Russian.

At the same time, Wikipedia entries that are important or interesting for people from many communities speaking different languages have articles in a variety of languages. For example, *Newton's law of universal gravitation* is a fundamental nature law and has descriptions in 30 languages. Interestingly, the Wikipedia entry about *Isaac Newton* who first formulated the law of universal gravitation and who is known all over the world has descriptions in 111 different languages.

However, even if a Wikipedia entry has articles in many languages, the information covered by these articles can differ substantially. The two main sources of differences are:

- the amount of the information covered by the Wikipedia articles (the length of the Wikipedia articles);
- the choice of the information covered by the Wikipedia articles.

For example, Wikipedia entry about *Isadora Duncan* has descriptions in 44 languages. The length of the descriptions about *Isadora Duncan* is different for every language: 127 sentences for the article in English; 77 - for French; 37 - for Russian, 1 - for Greek, etc. The question arises: whether a shorter article can be considered a summary of a longer article, or whether a shorter article might contain information that is either not covered in a longer article or contradicts the information in the longer article.

²http://en.wikipedia.org/wiki/List_of_Wikipedias

Wikipedia is changing constantly. All the quotes and examples from Wikipedia presented and analyzed in this paper were collected on February 10, 2009, between 14:00 and 21:00 PST.

Isadora Duncan was a American-born dancer who was very popular in Europe and was married to a Russian poet, *Sergey Esenin*. Certain amount of information facts (i.e., major biography dates) about *Isadora Duncan* are repeated in the articles in every language. However, shorter articles are not necessarily summaries of longer articles. For example, the article in Russian that is almost four time shorter than the articles in English, contains information that is not covered in the articles written in English. The same can be noted about articles in French and Spanish.

In this paper, we analyze the distribution of languages used in Wikipedia for the list of 48 people in the DUC 2004 biography generation task. We analyze, the number of languages that contain articles for each of the 48 DUC 2004 people. We also analyze the distribution of the lengths for the descriptions in different languages. We believe that this statistics is important for the understanding of the Wikipedia multilinguality nature and can be used by many NLP applications. Several NLP applications that can leverage this information are listed in Section 5.

4 Analysis of Wikipedia Multilinguality

In this paper, we propose a framework to quantify the multilinguality aspect of Wikipedia. In the current work we use a small portion of Wikipedia. Analyzing only a portion of Wikipedia allows us to compare in detail the multilinguality aspect for all the Wikipedia entries in our data set.

4.1 Data Set

For our analysis, we used the list of people created for the Task 5 of DUC 2004: biography generation task (48 people).³

First, we downloaded from Wikipedia all the articles in all the languages corresponding to each person from the DUC 2004 evaluation set. For our analysis we used Wikitext, the text that is used by Wikipedia authors and editors. Wikitext complies with the wiki markup language and can be processed by the Wikimedia content manager system into HTML which can then be viewed in a browser. This is the text that can be obtained through the

³<http://duc.nist.gov/duc2004/tasks.html/>

Wikipedia dumps.⁴ For our analysis we removed from the wikitext all the markup tags and tabular information (e.g., infoboxes and tables) and kept only plain text. There is no commonly accepted standard wikitext language, thus our final text had a certain amount of noise which, however, does not affect the conclusions drawn from our analysis.

For this work, for each Wikipedia entry (i.e., DUC 2004 person) we downloaded the corresponding descriptions in all the languages, including simple English, Esperanto, Latin, etc. To facilitate the comparison of descriptions written in different languages we used the Google machine translation system⁵ to translate the downloaded descriptions into English. The number of languages currently covered by the Google translation system (41 language) is smaller than the number of languages in which there exist Wikipedia articles (265 languages). However, we believe that using for cross-lingual analysis descriptions only in those languages that can be handled by the Google translation system does not affect the generality of our conclusions.

4.2 Data Processing Tools

After the Wikipedia descriptions for each person from the DUC 2004 set were collected and translated, we divided the description texts into sentences using the LingPipe sentence chunker (Alias-i, 2009). We apply sentence splitter only to the English language documents: either originally created in English or translated into English by the Google translation system.

4.3 Data Analysis

As mentioned in Section 1, the goal of the analysis described in this paper is to quantify the language diversity in Wikipedia entry descriptions.

We chose English as our reference and, for each DUC 2004 person, compared a description of this person in English against the descriptions of this person in other languages.

Language count: In Figure 1, we present information about descriptions in how many languages are created in Wikipedia for each person from the DUC 2004 set. All the people from the DUC 2004

⁴<http://download.wikimedia.org/>

⁵<http://translate.google.com/>

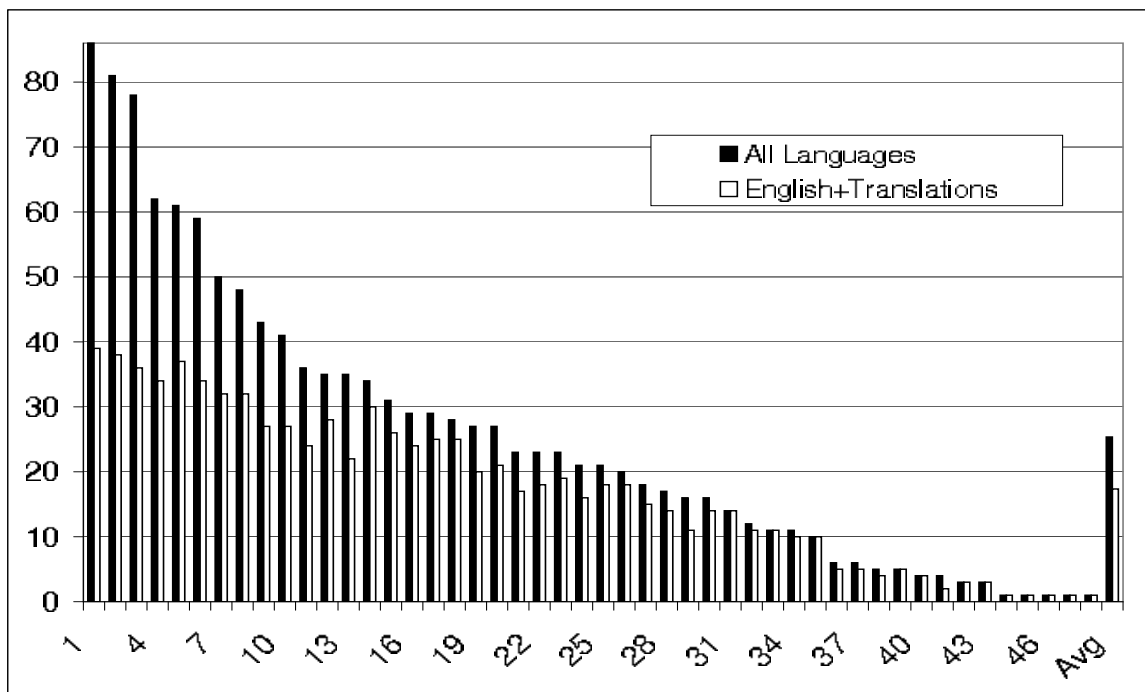


Figure 1: Number of languages for DUC 2004 people Wikipedia entries.

set have descriptions in English. The results in Figure 1 are presented in sorted order: from the Wikipedia entries with the largest number of descriptions (languages covered) to the Wikipedia entries with the smallest number of descriptions (languages covered). Five people from the DUC 2004 set have only one description (English). The person who has descriptions in the most number of languages for our data set is the former Secretary-General of the United Nations *Kofi Annan* (86 languages). Figure 1 also has information about descriptions in how many languages were translated into English (handled by the Google translation system).

Despite the fact that English is the language having descriptions for more Wikipedia entries than any other language, it does not always provide the greatest coverage for Wikipedia entries. To show this we analyzed the length of Wikipedia entry descriptions for the people from the DUC 2004 set. For our analysis, the length of a description is equal to the number of sentences in this description. To count the number of sentences in the uniform way for as many languages as possible we used translations of Wikipedia description from languages that are cur-

rently handled by the Google translation system into English. Those five people from the DUC 2004 set that have descriptions only in English are excluded from this analysis. Thus, in the data set for the next analysis we have 43 data points.

Sentence count: For every Wikipedia entry (person from the DUC 2004 set), we count the length of the descriptions originally created in English or translated into English by the Google translation system. In Figure 2, we present information about the length of the Wikipedia entity descriptions for English and for the language other than English with the maximum description length. The results in Figure 2 are presented in sorted order: from the Wikipedia entry with the maximal longest description in the language other than English to the Wikipedia entry with the minimal longest description in the language other than English for our data set. This sorted order does not correspond to the sorted order from Figure 1. It is interesting so see that the sorted order in Figure 2 does not correlate to the length distribution of English descriptions for our data set.

Obviously, the descriptions in English are not al-

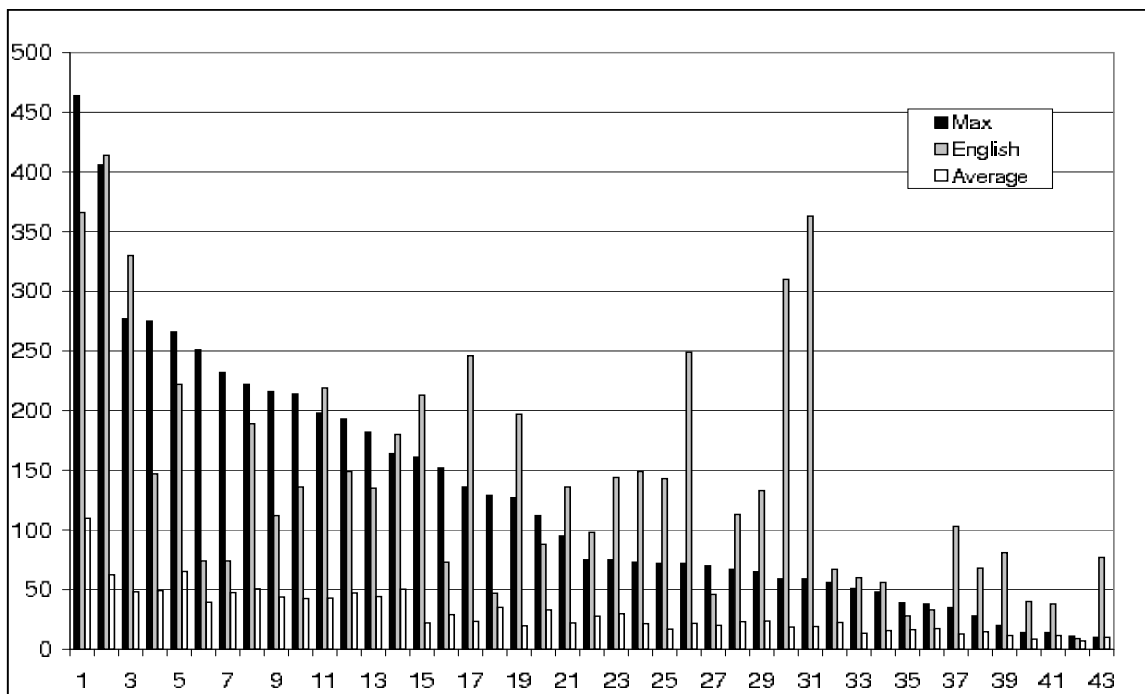


Figure 2: Number of sentences in the English description and the longest non-English description.

ways the longest ones. To be precise for 17 out of 43 people from the DUC 2004 set, the corresponding Wikipedia description in English was not the longest one. In several cases, the length of the description in English is several times shorter than the length of the longest (non-English) description. For example, the description of *Günter Grass* in German has 251 sentences while his description in English has 74 sentences.

It is safe to assume that longer descriptions have more information than shorter descriptions and 17 out of 43 English language descriptions of Wikipedia entries in our data set can be naturally extended with the information covered in the descriptions in other languages. Thus, multilingual Wikipedia gives a straight-forward way of extending Wikipedia entry descriptions.

It must be noted that the average length of Wikipedia descriptions (also presented on Figure 2) is very short. Thus, many descriptions for Wikipedia entries are quite short. The question arises how well the information covered in short descriptions corresponds to the information covered in long descriptions.

Correlation Analysis: In this paper, we present analysis for a small portion of Wikipedia. Currently, Wikipedia has more than more than 2,750,000 articles in English alone. Thus, the question arises whether our analysis can be used without loss of generality for the complete Wikipedia (i.e., all descriptions for all Wikipedia entries).⁶ To check this we analyzed the correspondence of how many Wikipedia entry descriptions are there for each language. For the Wikipedia subset corresponding to the people from the DUC 2004 set we simply counted how many Wikipedia entries have descriptions in each language. For the complete set of Wikipedia descriptions we used the Wikipedia size numbers from the *List of Wikipedias* page.⁷ After getting the Wikipedia size numbers we kept the data only for those languages that are used for descriptions of Wikipedia entries corresponding to the DUC 2004 people.

To compute correlation between these two lists of numbers we ranked numbers in each of these lists. The Rank (Spearman) Correlation Coefficient for

⁶It must be noted that the notion of *complete* Wikipedia is elusive as Wikipedia is changing constantly.

⁷http://en.wikipedia.org/wiki/List_of_Wikipedias

the above two ranked lists is equal to 0.763 which shows a high correlation between the two ranked lists. Thus, the preliminary analysis presented in work can be a good predictor for the descriptions' length distribution across descriptions in the complete multilingual Wikipedia.

5 Conclusions and Future Work

In this paper we presented a way of quantifying multilingual aspects of Wikipedia entry descriptions. We showed that despite the fact that English has descriptions for the most number of Wikipedia entries across all languages, English descriptions can not always be considered as the most detailed descriptions. We showed that for many Wikipedia entries, descriptions in the languages other than English are much longer than the corresponding descriptions in English.

Our estimation is that even though Wikipedia entry descriptions created in different languages are not identical, they are likely to contain information facts that appear in descriptions in many languages. One research direction that we are interested in pursuing is investigating whether the information repeated in multiple descriptions of a particular entry corresponds to the pyramid summarization model (Teufel and Halteren, 2004; Nenkova et al., 2007). In case of the positive answer to this question, multilingual Wikipedia can be used as a reliable corpus for learning summarization features.

Also, our preliminary analysis shows that Wikipedia entry descriptions might contain information that contradicts information presented in the entry descriptions in other languages. Even the choice of a title for a Wikipedia entry can provide interesting information. For example, the title for the Wikipedia entry about *Former Yugoslav Republic of Macedonia* in English, German, Italian, and many other languages uses the term *Republic of Macedonia* or simply *Macedonia*. However, Greece does not recognize this name, and thus, the title of the corresponding description in Greek has a complete formal name of the country: *Former Yugoslav Republic of Macedonia*.

Multilingual Wikipedia is full of information asymmetries. Studying information asymmetries in multilingual Wikipedia can boost research in new

information and contradiction detection. At the same time, information symmetries in multilingual Wikipedia can be used for learning summarization features.

References

- ACL. 2009. Workshop on building and using comparable corpora: from parallel to non-parallel corpora.
- Sisay Fissaha Adafre and Maarten de Rijke. 2006. Finding similar sentences across multiple languages in wikipedia. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, Workshop on New Text – Wikis and blogs and other dynamic text sources*, Trento, Italy, April.
- Eytan Adar, Michael Skinner, and Dan Weld. 2009. Information arbitrage in multi-lingual Wikipedia. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, Barcelona, Spain, February.
- David Ahn, Valentin Jijkoun, Gilad Mishne, Karin Müller, Maarten de Rijke, and Stefan Schlobach. 2005. Using Wikipedia at the TREC QA track. In *Proceedings of the Text REtrieval Conference (TREC 2004)*.
- Alias-i. 2009. Lingpipe 3.7.0. (accessed January 19, 2009). <http://alias-i.com/lingpipe>.
- Fadi Baidy, Julia Hirschberg, and Elena Filatova. 2008. An unsupervised approach to biography production using wikipedia. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-2008)*, Columbus, OH, USA, July.
- Regina Barzilaya and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, Toulouse, France, July.
- Davide Buscaldi and Paolo Rosso. 2006. Mining knowledge from wikipedia for the question answering task. In *Proceedings of The Fifth international Conference on Language Resources and Evaluation (LREC-2006)*, Genoa, Italy, May.
- CLEF. 2000. Cross-language evaluation forum (CLEF). <http://www.clef-campaign.org>.
- Sergio Ferrández, Antonio Toral, Óscar Ferrández, Antonio Ferrández, and Rafael Munoz. 2007. Applying Wikipedia's multilingual knowledge to cross-lingual question answering. *Lecture Notes in Computer Science (LNCS): Natural Language Processing and Information Systems*, 4592:352–363.
- Ulrich Germann. 2001. Aligned hansards of the 36th parliament of Canada. Website.

- <http://www.isi.edu/natural-language/download/hansard/>.
- Andrew Keen. 2007. *The Cult of the Amateur: How Today's Internet is Killing Our Culture*. Doubleday Business.
- Jeongwoo Ko, Teruko Mitamura, and Eric Nyberg. 2007. Language-independent probabilistic answer ranking for multilingual question answering. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-2007)*, Prague, Czech Republic, June.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Machine Translation Summit (MT-2005)*, Phuket Island, Thailand, September.
- LREC. 2008. Workshop on building and using comparable corpora.
- Lieve Macken, Julia Trushkina, and Lidia Rura. 2007. Dutch Parallel Corpus: MT corpus and translator's aid. In *Proceedings of the Eleventh Machine Translation Summit (MT-2007)*, pages 313–320, Copenhagen, Denmark, September.
- Bernardo Magnini, Simone Romagnoli, and Ro Vallin. 2003. Creating the DISEQuA corpus: A test set for multilingual question answering. In *Proceedings of the Cross-Lingual Evaluation Forum (CLEF-2003)*, Trondheim, Norway, August.
- Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The Pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing*, 4(2).
- Alexander Richman and Patrick Schone. 2008. Mining Wiki resources for multilingual named entity recognition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-2008)*, Columbus, OH, USA, July.
- Péter Schönhofen, András Benczúr, István Bíró, and Károly Csalogány. 2007. Performing cross-language retrieval with wikipedia. In *Proceedings of the Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary, September.
- Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of The Fifth international Conference on Language Resources and Evaluation (LREC-2006)*, Genoa, Italy, May.
- Simone Teufel and Hans Van Halteren. 2004. Evaluating information content by factoid analysis: Human annotation and stability. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics (ACL-2004)*, Barcelona, Spain, July.

Investigation in Statistical Language-Independent Approaches for Opinion Detection in English, Chinese and Japanese

Olena Zubaryeva

Institute of Informatics
University of Neuchâtel
Emile-Argand, 11, 2009 Switzerland
olena.zubaryeva@unine.ch

Jacques Savoy

Institute of Informatics
University of Neuchâtel
Emile-Argand, 11, 2009 Switzerland
jacques.savoy@unine.ch

Abstract

In this paper we present a new statistical approach to opinion detection and its' evaluation on the English, Chinese and Japanese corpora. Besides, the proposed method is compared with three baselines, namely Naïve Bayes classifier, a language model and an approach based on significant collocations. These models being language independent are improved with the use of language-dependent technique on the example of the English corpus. We show that our method almost always gives better performance compared to the considered baselines.

1 Introduction

The task of opinion mining has received attention from the research community and industry lately. The main reasons for extensive research in the area are the growth of user needs and companies' desire to analyze and exploit the user-generated content on the Web in the form of blogs and discussions. Thus, users want to search for opinions on various topics from products that they want to buy to opinions about events and well-known persons. A lot of businesses are interested in how their services are perceived by their customers. Therefore, the detection of subjectivity in the searched information may add the additional value to the interpretation of the results and their relevancy to the searched topic. The growth of user activi-

ty on the Web gives substantial amounts of data for these purposes.

In the context of globalization the possibility to provide search of opinionated information in different natural languages might be of great interest to organizations and communities around the world. Our goal is to design a fully automatic system capable of working in a language-independent manner. In order to compare our approach on different languages we chose English, traditional Chinese and Japanese corpora. As a further possibility to improve the effectiveness of the language independent methods we also consider the additional application of language dependent techniques specific to the particular natural language.

The related work in opinion detection is presented in Section 2. We describe our approach in detail with the three other baselines in Section 3. The fourth section describes language specific approach used for the English corpus. In Section 5 we present the evaluation of the three models using the NTCIR-6 and NTCIR-7 MOAT English, Chinese and Japanese test collections (Seki *et al.*, 2008). The main findings of our study and future research possibilities are discussed in the last sections.

2 Related Work

The focus of our work is to propose a general approach that can be easily deployed for different natural languages. This task of opinion detection is important in many areas of NLP such as question/answering, information retrieval, docu-

ment classification and summarization, and information filtering. There are numerous challenges when trying to solve the task of opinion detection. Some of them include the fact that the distinction between opinionated and factual could be denoted by a single word in the underlying text (e.g., “The iPhone price is \$600.” vs. “The iPhone price is high.”). Most importantly evaluating whether or not a given sentence conveys an opinion could be questionable when judged by different people. Further, the opinion classification can be done on different levels, from documents to clauses in the sentence.

We consider the opinion detection task on a sentence level. After retrieving the relevant sentences using any IR system we automatically classify a sentence according to two classes: opinionated and not opinionated (factual). When viewing an opinion-finding task as a classification task, it is usually considered as a supervised learning problem where a statistical model performs a learning task by analyzing a pool of labeled sentences. Two questions must therefore be solved, namely defining an effective classification algorithm and determining pertinent features that might effectively discriminate between opinionated and factual sentences. From this perspective, during the last TREC opinion-finding task (Macdonald *et al.*, 2008) and the last NTCIR-7 workshop (Seki *et al.*, 2008), a series of suggestions surfaced.

As the language-dependent approach various teams proposed using Levin defined verb categories (namely, characterize, declare, conjecture, admire, judge, assess, say, complain, advise) and their features (a verb corresponding to a given category occurring in the analyzed information item) that may be pertinent as a classification feature (Bloom *et al.*, 2007). However, words such as these cannot always work correctly as clues, for example with the word “said” in the two sentences “There were crowds and crowds of people at the concert, said Ann” and “There were more than 10,000 people at the concert, said Ann.” Both sentences contain the clue word “said” but only the first one contains an opinion on the target product. Turney (2002) suggested comparing the frequency of phrase co-occurrences with words predetermined by the sentiment lexicon. Specific to the opinion detection in Chinese language Ku *et al.* (2006) propose a dictionary-based approach for extraction and summarization. For the Japanese lan-

guage in the last NTCIR-6 and NTCIR-7 workshops the opinion finding methods included the use of supervised machine learning approaches with specific selection of certain parts-of-speech (POS) and sentence parts in the form of n -gram features to improve performance.

There has been a trend in applying language models for opinion detection task (Lavrenko, Croft, 2001). Pang & Lee (2004) propose the use of language models for sentiment analysis task and subjectivity extraction. Usually, language models are trained on the labeled data and as an output they give probabilities of classified tokens belonging to the class. Eguchi & Lavrenko (2006) propose the use of probabilistic language models for ranking the results not only by sentiment but also by the topic relevancy.

As an alternative other teams during the last TREC and NTCIR evaluation campaigns have suggested variations of Naïve Bayes classifier, language models and SVM, along with the use of such heuristics as word order, punctuation, sentence length, etc.

We might also mention OpinionFinder (Wilson *et al.*, 2005), a more complex system that performs subjectivity analyses to identify opinions as well as sentiments and other private states (speculations, dreams, etc.). This system is based on various classical computational linguistics components (tokenization, part-of-speech (POS) tagging (Toutanova & Manning, 2000) as well as classification tools. For example, a Naïve Bayes classifier (Witten & Frank, 2005) is used to distinguish between subjective and objective sentences. A rule-based system is included to identify both speech events (“said,” “according to”) and direct subjective expressions (“is happy,” “fears”) within a given sentence. Of course such learning system requires both a training set and a deeper knowledge of a given natural language (morphological components, syntactic analyses, semantic thesaurus).

The lack of enough training data for the learning-based systems is clearly a drawback. Moreover, it is difficult to objectively establish when a complex learning system has enough training data (and to objectively measure the amount of training data needed in a complex ML model).

3 Language Independent Approaches

In this section we propose our statistical approach for opinion detection as well as the description of the Naïve Bayes and language model (LM) baselines.

3.1 Logistic Model

Our system is based on two components: the extraction and weighting of useful features (limited to isolated words in this study) to allow an effective classification, and a classification scheme. First, we present the feature extraction approach in the Section 3.1.1. Next, we discuss our classification model. Sections 3.2 and 3.3 describe the chosen baselines.

3.1.1 Features Extraction

In order to determine the features that can help distinguishing between factual and opinionated documents, we have selected the tokens. As shown by Kilgarriff (2001), the selection of words (or in general features) in an effort to characterize a particular category is a difficult task. The goal is therefore to design a method capable of selecting terms that clearly belong to one of the classes. The approaches that use words and their frequencies or distributions are usually based on a contingency table (see Table 1).

	S	C-	
ω	a	b	$a+b$
not ω	c	d	$c+d$
	$a+c$	$b+d$	$n=a+b+c+d$

Table 1. Example of a contingency table.

In this table, the letter a represents the number of occurrences (tokens) of the word ω in the document set S (corresponding to a subset of the larger corpus C in the current study). The letter b denotes the number of tokens of the same word ω in the rest of the corpus (denoted C-) while $a+b$ is the total number of occurrences in the entire corpus (denoted C with $C=C \cup S$). Similarly, $a+c$ indicates the total number of tokens in S. The entire corpus C corresponds to the union of the subset S and C- that contains n tokens ($n = a+b+c+d$).

Based on the MLE (Maximum Likelihood Estimation) principle the values shown in a contingency table could be used to estimate various probabilities. For example we might calculate the probability of the occurrence of the word ω in the entire corpus C as $\Pr(\omega) = (a+b)/n$ or the probability of finding in C a word belonging to the set S as $\Pr(S) = (a+c)/n$.

Now to define the discrimination power a term ω , we suggest defining a weight attached to it according to Muller's method (Muller, 1992). We assume that the distribution of the number of tokens of the word ω follows a binomial distribution with the parameters p and n' . The parameter p represented the probability of drawing a word ω also denoted in the corpus C (or $\Pr(\omega)$) and could be estimated as $(a+b)/n$. If we repeat this drawing $n' = a+c$ times, we will have an estimate of the number of word ω included in the subset S by $\Pr(\omega) \cdot n'$. On the other hand, Table 1 gives also the number of observations of the word ω in S, and this value is denoted by a . A large difference between a and the product $\Pr(\omega) \cdot n'$ is clearly an indication that the presence of a occurrences of the term ω is not due by chance but corresponds to an intrinsic characteristic of the subset S compared to the subset C-.

In order to obtain a clear rule, we suggest computing the Z score attached to each word ω . If the mean of a binomial distribution is $\Pr(\omega) \cdot n'$, its variance is $n' \cdot \Pr(\omega) \cdot (1 - \Pr(\omega))$. These two elements are needed to compute the standard score as described in Equation 1.

$$Zscore(\omega) = \frac{a - n' \cdot \Pr(\omega)}{\sqrt{n' \cdot \Pr(\omega) \cdot (1 - \Pr(\omega))}} \quad (1)$$

As a decision rule we consider the words having a Z score between -2 and 2 as terms belonging to a common vocabulary, as compared to the reference corpus (as for example "will," "with," "many," "friend," or "forced" in our example). This threshold was chosen arbitrary. A word having a Z score > 2 would be considered as overused (e.g., "that," "should," "must," "not," or "government" in MOAT NTCIR-6 English corpus), while a Z score < -2 would be interpreted as an underused term (e.g., "police," "cell," "year," "died," or "agreeing"). The arbitrary threshold limit of 2 corresponds to the limit of the standard normal distribution, allowing us to find around 5% of the observa-

tions (around 2.5% less than -2 and 2.5% greater than 2). As shown in Figure 1, the difference between our arbitrary limit of 2 (drawn in solid line) and the limits delimiting the 2.5% of the observations (dotted line) are rather close.

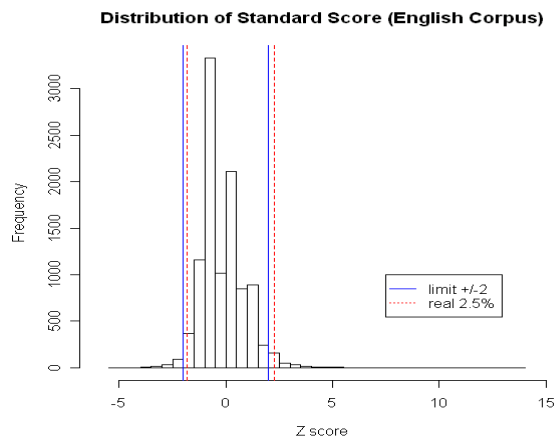


Figure 1. Distribution of the Z score (MOAT NTCIR-6 English corpus, opinionated).

Based on a training sample, we were able to compute the Z score for different words and retain only those having a large or small Z score value. Such a procedure is repeated for all classification categories (opinionated and factual). It is worth mentioning that such a general scheme may work with isolated words (as applied here) or n -gram (that could be a sequence of either characters or words), as well as with punctuations or other symbols (numbers, dollar signs), syntactic patterns (e.g., verb-adjective in comparative or superlative forms) or other features (presence of proper names, hyperlinks, etc.)

3.1.2 Classification Model

When our system needs to determine the opinionatedness of a sentence, we first represent this sentence as a set of words. For each word, we can then retrieve the Z scores for each category. If all Z scores for all words are judged as belonging to the general vocabulary, our classification procedure selects the default category. If not, we may increase the weight associated with the corresponding category (e.g., for the opinionated class if the underlying term is overused in this category).

Such a simple additive process could be viewed as a first classification scheme, selecting the class having the highest score after enumerating all words occurring in a sentence. This approach assumes that the word order does not have any im-

pact. We also assume that each sentence has a similar length.

For this model, we can define two variables, namely $SumOP$ indicating the sum of the Z score of terms overused in opinionated class (i.e. Z score > 2) and appearing in the input sentence. Similarly, we can define $SumNOOP$ for the other class. However, a large $SumOP$ value can be obtained by a single word or by a set of two (or more) words. Thus, it could be useful to consider also the number of words (features) that are overused (or underused) in a sentence. Therefore, we can define $\#OpOver$ indicated the number of terms in the evaluated sentence that tends to be overused in opinionated documents (i.e. Z score > 2) while $\#OpUnder$ indicated the number of terms that tends to be underused in the class of opinionated documents (i.e. Z score < -2). Similarly, we can define the variables $\#NoopOver$, $\#NoopUnder$, but for the non-opinionated category.

With these additional explanatory variables, we can compute the corresponding subjectivity score for each sentence as follows:

$$Op_score = \frac{\#OpOver}{\#OpOver + \#OpUnder} \quad (2)$$

$$Noop_score = \frac{\#NoopOver}{\#NoopOver + \#NoopUnder}$$

As a better way to combine different judgments we suggest following Le Calvé & Savoy (2000) and normalize the scores using the logistic regression. The logistic transformation $\pi(x)$ given by each logistic regression model is defined as:

$$\pi(x) = \frac{e^{\beta_0 + \sum_{i=1}^k \beta_i x_i}}{1 + e^{\beta_0 + \sum_{i=1}^k \beta_i x_i}} \quad (3)$$

where β_i are the coefficients obtained from the fitting, x_i are the variables, and k is the number of explanatory variables. These coefficients reflect the relative importance of each variable in the final score.

For each sentence, we can compute the $\pi(x)$ corresponding to the two possible categories and the final decision is simply to classify the sentence according to the max $\pi(x)$ value. This approach takes account of the fact that some explanatory variables may have more importance than other in assigning the correct category.

3.2 Naïve Bayes

For comparison with our logistic model we chose three baselines: Naïve Bayes and language model and finding significant collocations. Despite its simplicity Naïve Bayes classifier tends to perform relatively well for various text categorization problems (Witten, Frank, 2005). In accordance with our approach, we used word tokens as classification features for the English corpora. For the Chinese and Japanese languages overlapping bigram approach was used (Savoy, 2005). The training method estimates the relative frequency of the probability that the chosen feature belongs to a specific category using add-one smoothing technique.

3.3 Language Model (LM)

As a second baseline we use the classification based on the language model using overlapping n -gram sequences (n was set to 8) as suggested by Pang & Lee (2004, 2005) for the English language. Using the overlapping 4-gram sequence for the word “company”, we obtain: “comp”, “ompa”, “mpan”, etc. For the Chinese and Japanese corpora bigram approach was applied. As in Naïve Bayes, the language model gives the probability of the sentence belonging to a specific class. Working with relatively large n allows a lot of word tokens to be processed as is, at least for the English language.

3.4 Significant Collocations (SC)

Another promising approach among the supervised learning schemes is the use of collocations of two or more words or features (Manning & Schütze, 2000). This method allows classification of instances based on significant collocations learned from the labeled data. Some examples of the frequent collocations in the corpora would be “in the”, “of the”. The idea of the method is to find significant collocations (SC) that occur more in the opinionated corpus than in the non-opinionated one. In order to do so the model returns the collocations of two words for the English language based on the degree to which their counts in the opinionated corpus exceed their expected counts in the not opinionated one. As an example for the English opinionated corpus the following collocations were found: “are worried”, “pleaded guilty”, “eager to”, “expressed hope”. Clearly, overlooking the

list of new found collocations it is possible to judge their relevancy. However, it is not clear how to use this method with the Chinese and Japanese texts, since these languages do not have white space or other usual delimiters as in English. In order to solve the problem of feature selection we chose bigram indexing on the Chinese and Japanese corpora and searched for significant new collocations of bigrams.

4 Language Dependent Approach

As the language dependent technique to improve the obtained classification results we suggest the use of SentiWordNet for the English language (Esuli & Sebastiani, 2006). Since the vocabulary of words in SentiWordNet is quite limited it is not always clear how to combine the objectivity scores.

The SentiWordNet score was computed in the following way: to define the opinionated score of the sentence the sum of scores representing that the word belongs to opinionated category for each word in the sentence is calculated. The not opinionated score of the sentence is computed in the same way with the difference that it is divided by the number of words in the sentence. Thus, if opinionated score is more than not opinionated one, there is an opinion, otherwise not. This is a heuristic approach that intuitively takes account of the rationalization that there are more not opinionated words than opinionated in the sentence. At the same time the presence of opinionated word weighs more than the presence of the not opinionated ones. Especially, this approach seems to give good result.

5 Experiments

The experiment was carried out on the NTCIR-6 and NTCIR-7 English news corpora using 10-fold cross-validation model on a lenient evaluation standard as described in Seki et al. We do not question the construction and structure of opinions in this data set, since those questions were addressed at the NTCIR workshops. Using the Chinese and Japanese corpora we can verify the quality of the suggested language-independent approaches.

5.1 Feature Selection & Evaluation in English

For the evaluation of sentences in English, the assumption of isolated words (bag-of-words) previously stemmed was used by our system. The corpora are comprised of more than 13,400 sentences, 4,859 (36.3%) of which are opinionated. As the evaluation metrics precision, recall and F_1 -measure were used based on gold standard evaluation provided by NTCIR workshops (Seki *et al.*, 2008). The precision and recall are weighted equally in our experiment but it should be recognized that based on the system's needs and focus there could be more accent on precision or recall.

Model	Precision	Recall	F_1 -measure
Logistic model	0.583	0.508	0.543
Naïve Bayes	0.415	0.364	0.388
LM	0.350	0.339	0.343
SC	0.979	0.360	0.527

Table 2. Evaluation results of 10-fold cross-validation on NTCIR-6 and NTCIR-7 English corpora.

Comparing the results in Table 2 to the baselines of the Naïve Bayes classifier and LM evaluated on the same training and testing sets, we see that logistic model outperforms the baselines. In our opinion, this is due to the use of more explanatory variables that better discriminate between opinionated and factual sentences.

The use of language dependent techniques on the other hand might further improve the results. Especially, this seems promising observing the results when using the SentiWordNet on the English corpus. In Table 3 one can see that the first three models show improvement. Specifically, the precision of the logistic model increased from 0.583 to 0.766 (by 31.4%).

Model	Precision	Recall	F_1 -measure
Logistic model	0.766	0.488	0.597
Naïve Bayes	0.667	0.486	0.562
LM	0.611	0.474	0.534
SC	0.979	0.420	0.588

Table 3. Evaluation results of 10-fold cross-validation on NTCIR-6 and NTCIR-7 English corpora with SentiWordNet.

When considering the F_1 -measure, the impact of the language-dependent approach shows 9% of improvement, from 0.543 to 0.597.

The way that we incorporated the scores provided by SentiWordNet was done with the help of linear combination and normalization of scores for each of the models.

5.2 Feature Selection & Evaluation in Chinese

We have assumed until now that words can be extracted from a sentence in order to define the needed features used to determine if the underlying information item conveys an opinion or not. Working with the Chinese language this assumption does no longer hold. Therefore, we need to determine indexing units by either applying an automating segmentation approach (based on either a morphological (e.g., CSeg&Tag) or a statistical method (Murata & Isahara, 2003)) or considering n -gram indexing approach (unigram or bigram, for example). Finally we may also consider a combination of both n -gram and word-based indexing strategies.

Based on the work of Savoy, 2005 we experimented with overlapping bigram and trigram indexing schemes for Chinese. The experimental results show that bigram indexing outperforms trigram on all three considered statistical methods. Therefore, as features for Chinese we used overlapping bigrams.

The NTCIR-6 and NTCIR-7 Chinese corpora consisted of more than 14,507 sentences, 9960 (68.7%) of which are opinionated. The results of all three statistical models performed on the Chinese corpora are presented in Table 4.

Model	Precision	Recall	F_1 -measure
Logistic model	0.943	0.730	0.823
Naïve Bayes	0.729	0.538	0.619
LM	0.581	0.634	0.606
SC	0.313	0.898	0.464

Table 4. Evaluation results of 10-fold cross-validation on NTCIR-6 and NTCIR-7 Chinese corpora.

From the results in Table 4 we clearly see that our approach gives better performance and confirms the results presented in Tables 2 and 3. The significant improvement in scores could be due to the fact that Chinese corpus contains more opinionated sentences in relevance to not opinionated once. Thus, the training set for opinionated classi-

fication was much larger compared to the English language. This proves the relevance of more training data for the learning-based systems. But the direct comparison with the results on the English corpus is not possible.

5.3 Feature Selection & Evaluation in Japanese

As with the Chinese language we face the same challenges in feature definition for the Japanese language. After experimenting with bigram and trigram we chose bigram strategy for indexing and feature selection.

The NTCIR-6 and NTCIR-7 Japanese corpora consisted of more than 11,100 sentences with 4,622 opinionated sentences (representing 41.6% of the corpus). The results of the statistical models are shown in Table 5.

Model	Precision	Recall	F-measure
Logistic model	0.527	0.761	0.623
Naïve Bayes	0.565	0.570	0.567
LM	0.657	0.667	0.662
SC	0.663	0.856	0.747

Table 5. Evaluation results of 10-fold cross-validation on NTCIR-6 and NTCIR-7 Japanese corpora.

From the results we can see that the significant collocations model outperforms the others. This could be due to the fewer number of opinionated sentences compared to the Chinese or English corpora. This tends to indicate the necessity of an extensive training data for the logistic model in order to provide reliable opinion estimates.

6 Future Work and Conclusion

In this paper we presented our language-independent approach based on using *Z* scores and the logistic model to identify those terms that adequately characterize subsets of the corpus belonging to opinionated or non-opinionated classes. In this selection, we focused only on the statistical aspect (distribution difference) of words or bigrams. Our approach was compared to the three baselines, namely Naïve Bayes classifier, language model and an approach based on finding significant collocations. We have also demonstrated on the English corpora how we can use the language dependent techniques to identify the possibility of opinion ex-

pressed in the sentences that otherwise were classified as not opinionated by the system. The use of SentiWordNet (Esuli & Sebastiani, 2006) in combination with our methods yields better results for the English language.

This study was limited to isolated words in English corpus but in further research we could easily consider longer word sequences to include both noun and verb phrases. The most useful terms would also then be added to the query to improve the rank of opinionated documents. As another approach, we could use the evaluation of co-occurrence terms of pronouns “I” and “you” mainly with verbs (e.g., “believe,” “feel,” “think,” “hate”) using part of speech tagging techniques in order to boost the rank of retrieved items.

Using freely available POS taggers, we could take POS information into account (Toutanova & Manning, 2004) and hopefully develop a better classifier. For example, the presence of proper names and their frequency or distribution might help us classify a document as being opinionated or not. The presence of adjectives and adverbs, together with their superlative (e.g., best, most) or comparative (e.g., greater, more) forms could also be useful hints regarding the presence of opinionated versus factual information.

Acknowledgments

We would like to thank the MOAT task organizers at NTCIR-7 for their valuable work.

References

- Bloom, K., Stein, S., & Argamon, S. 2007. *Appraisal extraction for news opinion analysis at NTCIR-6*. Proceedings NTCIR-6, NII, Tokyo, pp. 279-289.
- Eguchi, K., Lavrenko, V. 2006. *Sentiment retrieval using generative models*. Proceedings of EMNLP, Sydney, pp. 345-354.
- Esuli, A., Sebastiani, F. 2006. *SentiWordNet: A publicly available lexical resource for opinion mining*. Proceedings LREC’06, Genoa.
- Kilgarriff, A. 2001. *Comparing corpora*. International Journal of Corpus Linguistics, 6(1):97-133.
- Ku, L.-W., Liang, Y.-T., Chen, H.-H. 2006. *Opinion extraction, summarization and tracking in news and blog corpora*. Proceedings of AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs, pp. 100-107.
- Lavrenko, V., Croft, W.B. 2001. *Relevance-based language models*. SIGIR, New Orleans, LA, pp. 120-127.

- Le Calvé, A., Savoy, J. 2000. *Database merging strategy based on logistic regression*. Information Processing & Management, 36(3):341-359.
- Macdonald, C., Ounis, I., & Soboroff, I. 2008. *Overview of the TREC-2007 blog track*. In Proceedings TREC-2007, NIST Publication #500-274, pp. 1-13.
- Manning, C. D., Schütze, H. 2000. *Foundations of Statistical Natural Language Processing*. MIT Press.
- Muller, C. 1992. *Principes et méthodes de statistique lexicale*. Champion, Paris.
- Murata, M., Ma, Q., & Isahara, H. 2003. *Applying multiple characteristics and techniques to obtain high levels of performance in information retrieval*. Proceedings of NTCIR-3, NII, Tokyo.
- Pang, B., Lee, L. 2004. *A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts*. Proceedings of ACL, Barcelona, pp. 271-278.
- Pang, B., Lee, L. 2005. *Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales*. In Proceedings of the Association for Computational Linguistics (ACL), pp. 115-124.
- Savoy, J. 2005. *Comparative study of monolingual search models for use with asian languages*. ACM Transactions on Asian Language Information Processing, 4(2):163-189.
- Seki, Y., Evans, D. K., Ku, L.-W., Sun, L., Chen, H.-H., & Kando, N. 2008. *Overview of multilingual opinion analysis task at NTCIR-7*. Proceedings NTCIR-7, NII, Tokyo, pp. 185-203.
- Toutanova, K., & Manning, C. 2000. *Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagging*. Proceedings EMNLP / VLC-2000, Hong Kong, pp. 63-70.
- Turney, P. 2002. *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*. Proceedings of the ACL, Philadelphia (PA), pp. 417-424.
- Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., & Patwardhan, S., 2005. *OpinionFinder: A system for subjectivity analysis*. Proceedings HLT/EMNLP, Vancouver (BC), pp. 34-35.
- Witten, I.A., & Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco (CA).

Sentence Position revisited: A robust light-weight Update Summarization ‘baseline’ Algorithm

Rahul Katragadda

rahul.k@research.iiit.ac.in

Prasad Pingali

pvpvr@iiit.ac.in

Vasudeva Varma

vv@iiit.ac.in

Language Technologies Research Center
IIIT Hyderabad

Abstract

In this paper, we describe a sentence position based summarizer that is built based on a sentence position policy, created from the evaluation testbed of recent summarization tasks at Document Understanding Conferences (DUC). We show that the summarizer thus built is able to outperform most systems participating in task focused summarization evaluations at Text Analysis Conferences (TAC) 2008. Our experiments also show that such a method would perform better at producing short summaries (upto 100 words) than longer summaries. Further, we discuss the baselines traditionally used for summarization evaluation and suggest the revival of an old baseline to suit the current summarization task at TAC: the Update Summarization task.

1 Introduction

Document summarization received a lot of attention since an early work by Luhn (1958). Statistical information derived from word frequency and distribution was used by the machine to compute a relative measure of significance, first for individual words and then for sentences. Later, Edmundson (1969) introduced four clues for identifying significant words (topics) in a text. Among them *title* and *location* are related to position methods, while the other two are *presence of cue words* and *high frequency content words*. Edmundson assigned positive weights to sentences according to their ordinal position in the text, giving more weight to the first sentence in the first paragraph and last sentence in the last paragraph.

Position of a sentence in a document or the position of a word in a sentence give good clues towards importance of the sentence or word respectively. Such features are called locational features, and a *sentence position* feature deals with presence of key sentences at specific locations in the text. Sentence Position has been well studied in summarization research since its inception, early in Edmundson’s work (1969). Earlier, Baxendale (1958) investigated a sample of 200 paragraphs to determine where the important words are most likely to be found. He concluded that in 85% of the paragraphs, the first sentence was a topic sentence and in 7% of the paragraphs, the final one.

Recent advances in machine learning have been adapted to summarization problem through the years and locational features have been consistently used to identify salience of a sentence. Some representative work in ‘learning’ sentence extraction would include training a binary classifier (Kupiec et al., 1995), training a Markov model (Conroy et al., 2004), training a CRF (Shen et al., 2007), and learning pairwise-ranking of sentences (Toutanova et al., 2007).

In recent years, at the Document Understanding Conferences (DUC¹), Text Summarization research evolved through task focused evaluations ranging from ‘*generic single-document summarization*’ to ‘*query-focused multi-document summarization (QFMDS)*’. The QFMDS task models the real-world complex question answering task wherein, given a topic and a set of 25 relevant documents, the

¹<http://duc.nist.gov/>

task is to synthesize a fluent, well-organized 250-word summary of the documents that answers the question(s) in the topic statement. Recent focus in the community has been towards *query-focused update-summarization* task at DUC and the Text Analysis Conference (TAC²). The update task was to produce short (~100 words) multi-document update summaries of newswire articles under the assumption that the user has already read a set of earlier articles. The purpose of each update summary will be to inform the reader of new information about a particular topic.

The rest of the paper is organized as follows. In Section 2, we describe a Sub-optimal Position Policy (SPP) based on Pyramid Annotated Data, then we derive a simple algorithm for summarization based on the SPP in Section 3, and show evaluation results. Next, in Section 4, we explain the current baselines and evaluation for Multi-Document Summarization and finally in Section 5, we discuss the need for an older baseline in the current context of the short summary task of update summarization.

2 Sub-Optimal Sentence Position Policy

Given a large text collection and a way to approximate the relevance for a reasonably large subset of sentences, we could identify significant positional attributes for the genre of the collection. Our experiments are based on the work described in (Lin and Hovy, 1997), whose experiments using the Ziff-Davis corpus gave great insights on the selective power of the position method.

2.1 Sentence Position Yield and Optimal Position Policy (OPP)

Lin and Hovy (1997) provide an empirical validation for the position hypothesis. They describe a method of deriving an Optimal Position Policy for a collection of texts within a genre, as long as a small set of topic keywords is defined for each text. They defined *sentence yield* (strength of relevance) of a sentence based on the mention of topic keywords in the sentence.

The *positional yield* is defined as the average *sentence yield* for that position in the document. They

²<http://www.nist.gov/tac/>

computed the yield of each sentence position in each document by counting the number of different keywords contained in the respective sentence in each document, and averaging over all documents. An Optimal Position Policy (OPP) is derived based on the decreasing values of *positional yield*.

Their experiments grounded on the assumption that abstract is an ideal representation of central topic(s) of a text. For their evaluations, they used the abstract to compare whether the sentences found based on their Optimal Position Policy are indeed a good selection. They used precision-recall measures to establish those findings.

At our disposal we had data from pyramid evaluations that provided sentences and their mapping to any content units in the gold standard summaries. The annotations in the data provide a unique property that each sentence can derive for itself a score for relevance.

2.2 Documents

There are a wide variety of document types across genre. In our case of newswire collection we have identified two primary types of documents: *small document* and *large document*. This distinction is made based on the total sentences in the document. All documents that have the number of sentences above a threshold should be considered large. We experimented on thresholds varying from 10 to 35 sentences and figured out that documents' distribution into the two categories was acceptable when threshold-ed at 20 sentences. This decision is also well supported by the fact that the last sentences of a document were more important than the others in the middle (Baxendale, 1958).

Sentence Position Yield (SPY) is obtained separately for both types of documents. For a small document, sentence positions have values from 1 through 20. Meanwhile, for a large document we compute SPY for position 1 through 20, then the last 15 sentences labeled 136 through 150 and '*any other sentence*' is labeled 100. It can be seen in figure 3 that sentences that do not come from leading or trailing part of large documents do not contribute much content to the summaries.

```

<document name="APW20000824.0204">
<line>
A lawyer who specializes in bankrupting hate groups is going after the Aryan Nations, whose compound in the Idaho
woods has served as a clubhouse for some of America's most violent racists.</line>
<line>
In a lawsuit that goes to trial Monday, attorney Morris Dees of the Southern Poverty Law Center is representing a
mother and son who were attacked by security guards for the white supremacist group.<annotation scu-count="1" sum-
count="8" sums="13,14,15,23,24,29,30,9">
<scu uid="24" label="SPLC takes legal action against civil rights abuses" weight="3"/>
</annotation>
</line>
<line>
The victims are suing the Aryan Nations and founder Richard Butler.<annotation scu-count="0" sum-count="1" sums="2
9"/>
</line>

```

Figure 1: A sample mapping of SCU annotation to source document sentences. An excerpt from mapping of topic D0701A of DUC 2007 QF-MDS task.

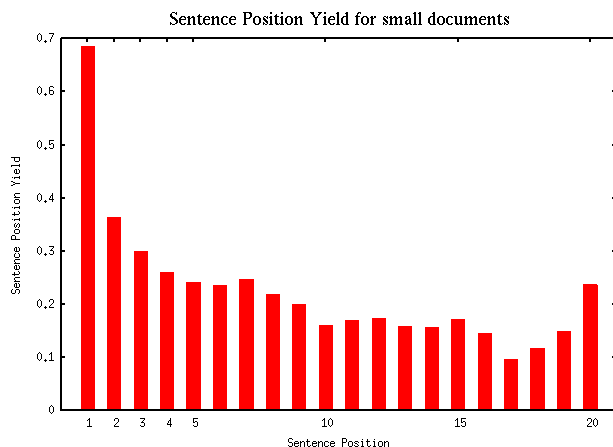


Figure 2: Sentence Position Yield for small documents.

2.3 Pyramid Data

Summary content units, referred as SCUs hereafter, are semantically motivated, sub-sentential units that are variable in length but not bigger than a sentential clause. SCUs emerge from annotation of a collection of human summaries for the same input. They are identified by noting information that is repeated across summaries, whether the repetition is as small as a modifier of a noun phrase or as large as a clause. The weight an SCU obtains is directly proportional to the number of reference summaries that support that piece of information. The evaluation method that is based on overlapping SCUs in human and automatic summaries is described in the Pyramid method (Nenkova et al., 2007).

The University of Ottawa has organized the pyramid annotation data such that for some of the sentences in the original document collection (those

that were picked by systems participating in pyramid evaluation), a list of corresponding content units is known (Copeck et al., 2006). We used this data to identify locations in a document from where most sentences were being picked, and which of those locations were being most content responsive to the query.

A sample of SCU mapping is shown in figure 1. Three sentences are seen in the figure among which two have been annotated with system IDs and SCU weights wherever applicable. The first sentence has not been picked by any of the summarizers participating in Pyramid Evaluations, hence it is unknown if the sentence would have contributed to any SCU. The second sentence was picked by 8 summarizers and that sentence contributed to an SCU of weight 3. The third sentence in the example was picked by one summarizer, however, it did not contribute to any SCU. This example shows all the three types of sentences available in the corpus: unknown samples, positive samples and negative samples.

For each SCU, a weight is associated in pyramid annotations. Thus a sentential score could be defined as sum of weights of all the contributing SCUs of the sentence. For an unknown sample and a negative sample, sentential score is 0. For example, in the second sentence in figure 1 the score is 3, contributed by a single SCU. While the same for the first and third sentences is 0.

For each sentence position the sentential score is averaged over all documents, which we call Sentence Position Yield. SPY for small and large documents is shown in figures 2 and 3. Based on these values for various positions, a simple Position Pol-

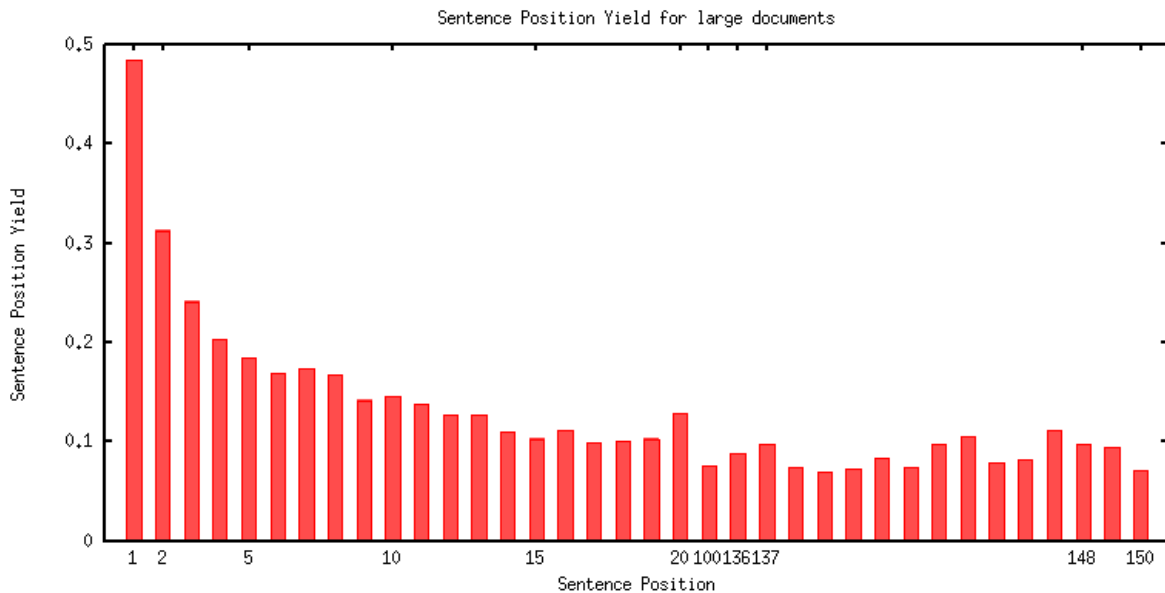


Figure 3: Sentence Position Yield for large documents

icy was framed as shown below. A position policy is an ordered set consisting of elements in the order of most importance. Within a subset, each sub-element is equally important and treated likewise.

$$\{s1, S1, \{s2, S2, s3\}, \{S3, s4, s5, s6, s7, s8, s20\}, \{S4, s9\} \dots \}$$

In the above position policy, sentences from small documents and large documents are represented by s_i and S_j respectively.

The position policy described above provides an ordering of ranked sentence positions based on a very accurate ‘relevance’ annotations on sentences. However, there is a large subset of sentences that are not annotated with either positive or negative relevance judgment. Hence, the policy derived is based on a high-precision low-recall corpus³ for sentence relevance. If all the sentences were annotated with such judgements, the policy could have been different. For this reason we call the above derived policy, a Sub-optimal Position Policy (SPP).

3 SPP as an algorithm

The goal of creating a position policy was to identify its effectiveness as a summarization algorithm. The

³DUC 2005 and 2006 data has been used for learning the SPP. In further experiments in section 3, DUC 2007 and TAC 2008 data have been used as test data.

above simple heuristic was easily incorporated as an algorithm based on simple scoring for each distinct set in the policy. For instance, based on the policy above, all $s1$ get the highest weight followed by next best weight to all $S1$ and so on.

As it can be observed, only the first sentence of each document could end up comprising the summary. This is okay, till we don’t get redundant information in the summary. Hence we also used a simple unigram match based redundancy measure that doesn’t allow a sentence if it matches any of the already selected sentences in at least 40% of content words in it. We also dis-allow sentences greater than 25 content words.

We applied the above algorithm to generate multi-document summaries for various tasks. We have applied it to Query-Focused Multi-Document Summarization (QF-MDS) task of DUC 2007 and Query-Focused Update Summarization task of TAC 2008.

3.1 Query-Focused Multi-Document Summarization

The *query-focused multi-document summarization* task at DUC models the real world complex question answering task. Given a topic and a set of 25 relevant documents, this task is to synthesize a fluent, well-organized 250 word summary of the documents that answers the question(s) in the topic state-

ment/narration.

The summaries from the above algorithm for the QF-MDS were evaluated based on ROUGE metrics (Lin, 2004). The average⁴ recall scores are reported for ROUGE-2 and ROUGE-SU4 in Table 1. Also reported are the performance of the top performing system and the official baseline(s). This algorithm performed worse than most systems participating in the task that year and performed better⁵ than only the ‘first x words’ baseline and 3 other systems.

system	ROUGE-2	ROUGE-SU4
‘first x words’ baseline	0.06039	0.10507
‘generic’ baseline	0.09382	0.14641
<i>SPP algorithm</i>	0.06913	0.12492
system 15 (top system)	0.12448	0.17711

Table 1: ROUGE 2, SU4 Recall scores for two baselines, the *SPP algorithm* and a top performing system at Query-Focused Multi-Document Summarization task, DUC 2007.

3.2 Update Summarization Task

The update summarization task is to produce short (~100 words) multi-document update summaries of newswire articles under the assumption that the user has already read a set of earlier articles. The initial document set is called *cluster A* and the next set of articles are called *cluster B*. For cluster A, a query-focused multi-document summary is expected. The purpose of each ‘update summary’ (summary of *cluster B*) will be to inform the reader of new information about a particular topic. Summaries from the above algorithm for the Query Focused Update Summarization task were evaluated based on ROUGE metrics. This algorithm performed surprisingly better at this task when compared to QF-MDS. The rouge scores suggest that this algorithm is well above the median for cluster A and among the top 5 systems for cluster B.

It must be noted that consistent performance across clusters (both A and B) shows the *robustness* of the ‘*SPP algorithm*’ at the update summarization task. Also, it is evident that such an algorithm is computationally simple and *light-weight*.

⁴Averaged over all the 45 topics of DUC 2007 dataset.

⁵Better in a statistical sense, based on 95% confidence intervals of the two systems’ evaluation based on ROUGE-2.

These surprisingly high scores on ROUGE metrics prompted us to evaluate the summaries based on Pyramid Evaluation (Nenkova et al., 2007). Pyramid evaluation provides a more semantic approach to evaluation of content based on SCUs as discussed in Section 2.3. The average⁶ modified pyramid scores of cluster A and cluster B summaries is shown in Table 2, along with the average recall scores for ROUGE-2, ROUGE-SU4 scores. The pyramid evaluation⁷ suggests that this algorithm performs better than all other automated systems at TAC 2008. Table 3 shows the average performance (across clusters) of ‘first x words’ baseline, SPP algorithm and two top performing systems (System ID=43 and ID=11). System 43 was adjudged best system based on ROUGE metrics, and system 11 was top performer based on pyramid evaluations at TAC 2008.

	ROUGE-2	ROUGE-SU4	pyramid
cluster A	0.08987	0.1213	0.3432
cluster B	0.09319	0.1283	0.3576

Table 2: Cluster wise ROUGE 2, SU4 Recall scores and modified Pyramid Scores for SPP algorithm at the Update Summarization task.

3.3 Discussion

It is interesting to observe that the algorithm that performs very poorly at QF-MDS, does very well in the Update Summarization task. A possible explanation for such behavior could be based on summary length. For a 250 word summary in the QF-MDS task, human summaries might provide a descriptive answer to the query that includes information nuggets accompanied by background information. Indeed, it has been earlier reported that humans appreciate receiving more information than just the answer to the query, whenever possible (Lin et al., 2003; Bosma, 2005).

Whereas, in the case of Update Summarization task the summary length is only 100 words. In such a short length humans need to trade-off between answer sentences and supporting sentences, and usually answers are preferred. And since our method

⁶Averaged over all the 48 topics of TAC 2008 dataset.

⁷Pyramid Annotation were done by a volunteer who also volunteered for annotations during DUC 2007.

system	ROUGE-2	ROUGE-SU4	pyramid
‘first x words’ baseline	0.05896	0.09327	0.166
SPP algorithm	0.09153	0.1245	0.3504
System 43 (top in ROUGE)	0.10395	0.13646	0.289
System 11 (top in pyramid)	0.08858	0.12484	0.336

Table 3: Average ROUGE 2, SU4 Recall scores and modified Pyramid Scores for baseline, SPP algorithm and two top performing systems at TAC 2008.

identifies sentences that are known to be contributing towards the needed answers, it performs better at the shorter version of the task.

Another possible explanation is that as a shorter summary length is required, the task of choosing the most important information becomes more difficult and no approach works well consistently. Also, it has often been noted that this baseline is indeed quite strong for this genre, due to the journalistic convention for putting the most important part of an article in the initial paragraphs.

4 Baselines in Summarization Tasks

Over the years, as summarization research followed trends from *generic single-document summarization*, to *generic multi-document summarization*, to *focused multi-document summarization* there were two major baselines that stayed throughout the evaluations. Those two baselines are:

1. First N words of the document (or of the most recent document).
2. First sentence from each document in chronological order until the length requirement is reached.

The first baseline was in place ever since the first evaluation of *generic single document summarization* took place in DUC 2001. For multi-document summarization, first N words of the most recent document (chronologically) was chosen as the baseline 1. In the recent summarization evaluations at Text Analysis Conference (TAC 2008), where update summarization was evaluated; baseline 1 still persists. This baseline performs pretty poorly at content evaluations based on all manual and automatic metrics. However, since it doesn’t disturb the original flow and ordering of a document, linguistically these summaries are the best. Indeed it outperforms all the automated systems based on linguistic quality evaluations.

The second baseline had been used occasionally with multi-document summarization from 2001 to 2004 with both generic multi-document summarization and focused multi-document summarization. In 2001 only one system significantly outperformed the baseline 2 (Nenkova, 2005). In 2003 QF-MDS however, only one system outperformed the baseline 2 above, while in 2004 at the same task, no system significantly outperforms the baseline. This baseline as can be seen, over the years has been pretty much untouched by systems based on content evaluation. However, the linguistic aspects of summary quality would be compromised in such a summary.

Currently, for the Update Summarization task at TAC 2008, NIST’s baseline is the baseline 1 (‘first x words’ baseline). And all systems (except one) perform better than the baseline in all forms of content evaluation. Since the task is to generate 100 word summaries (short summaries), based on past experiences, there is no doubt that baseline 2 would perform well.

It is interesting to observe that baseline 2 is a close approximation to the ‘SPP algorithm’ described in this paper. There are two main differences that we draw between ‘baseline 2’ and SPP algorithm. First, ‘baseline 2’ picks only the first sentence in each document, while ‘SPP algorithm’ could pick other sentences in an order described by the position policy. Second, ‘baseline 2’ puts no restriction on redundancy, thus due to journalistic conventions entire summary might be comprised of the same ‘information nuggets’, wasting the minimal real-estate available (~100 words). On the other hand, in our ‘SPP algorithm’ we consider a simple unigram-overlap measure to identify redundant information in sentence pairs that avoids redundant nuggets in the final summary.

5 Discussion and Conclusion

Baselines 1 and 2 mentioned above, could together act as a balancing mechanism to compare for linguistic quality and responsive content in the summary. The availability of a stronger content responsive summary as a baseline would enable steady progress in the field. While all the linguistically motivated systems would compare themselves with baseline 1, the summary content motivated systems would compare with the stronger baseline 2 and get better than it.

Over the years to come, the usage of ‘baseline 1’ doesn’t help in understanding whether there has been significant improvement in the field. This is because almost every simple algorithm beats the baseline performance. Having a better baseline, like the one based on the position hypothesis, would raise the bar for systems participating in coming years, and tracking progress of the field over the years is easier.

In this paper, we derived a method to identify a ‘sub-optimal position policy’ based on pyramid annotation data, that were previously unavailable. We also distinguish small and large documents to obtain the position policy. We described the Sub-optimal Sentence Position Policy (SPP) based on pyramid annotation data and implemented the SPP as an algorithm to show that a position policy thus formed is a good representative of the genre and thus performs way above median performance. We further describe the baselines used in summarization evaluation and discuss the need to bring back baseline 2 (or the ‘SPP algorithm’) as an official baseline for *update summarization* task.

Ultimately, as Lin and Hovy (1997) suggest, the position method can only take us certain distance. It has a limited power of resolution (the sentence) and its limited method of identification (the position in a text). Which is why we intend to use it as a baseline. Currently, as we can see the algorithm generates a *generic* summary, it doesn’t consider the topic or query to generate a *query-focused* summary. In future we plan to extend the SPP algorithm with some basic method for bringing in relevance.

References

P. B. Baxendale. 1958. Machine-made index for technical literature – an experiment. *IBM Journal of Re-*

- search and Development*, 2(Non-topical Issue).
- Wauter Bosma. 2005. Extending answers using discourse structures. In Horacio Saggion and J. L. Minel, editors, *RANLP workshop on Crossing Barriers in Text summarization Research*, pages 2–9. Incoma Ltd.
- John M. Conroy, Judith D. Schlesinger, Jade Goldstein, and Dianne P. O’leary. 2004. Left-brain/right-brain multi-document summarization. In *the proceedings of Document Understanding Conference (DUC) 2004*.
- Terry Copeck, D Inkpen, Anna Kazantseva, A Kennedy, D Kipp, Vivi Nastase, and Stan Szpakowicz. 2006. Leveraging duc. In *proceedings of DUC 2006*.
- H. P. Edmundson. 1969. New methods in automatic extracting. In *Journal of the ACM*, volume 16, pages 264–285. ACM.
- Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *the proceedings of ACM SIGIR’95*, pages 68–73. ACM.
- Chin-Yew Lin and Eduard Hovy. 1997. Identifying topics by position. In *Proceedings of the fifth conference on Applied natural language processing*, pages 283–290. ACL.
- Jimmy Lin, Dennis Quan, Vineet Sinha, Karun Bakshi, David Huynh, Boris Katz, and David R. Karger. 2003. The role of context in question answering systems. In *the proceedings of CHI’04*. ACM.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *the proceedings of ACL Workshop on Text Summarization Branches Out*. ACL.
- H.P. Luhn. 1958. The automatic creation of literature abstracts. In *IBM Journal of Research and Development*, Vol. 2, No. 2, pp. 159-165, April 1958.
- Ani Nenkova, Rebecca Passonneau, and Kathleen Mckeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. In *ACM Trans. Speech Lang. Process.*, volume 4, New York, NY, USA. ACM.
- Ani Nenkova. 2005. Automatic text summarization of newswire: Lessons learned from the document understanding conference. In Manuela M. Veloso and Subbarao Kambhampati, editors, *AAAI*, pages 1436–1441. AAAI Press / The MIT Press.
- Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. 2007. Document summarization using conditional random fields. In *the proceedings of IJCAI ’07.*, pages 2862–2867. IJCAI.
- Kristina Toutanova, Chris Brockett, Michael Gamon, Jagadeesh Jagarlamundi, Hisami Suzuki, and Lucy Vanderwende. 2007. The pythy summarization system: Microsoft research at duc 2007. In *the proceedings of Document Understanding Conference 2007*.

An Approach to Text Summarization

Sankar K

AU-KBC Research Centre
MIT Campus, Anna University
Chennai- 44.
sankar@au-kbc.org

Sobha L

AU-KBC Research Centre
MIT Campus, Anna University
Chennai- 44.
sobha@au-kbc.org

Abstract

We propose an efficient text summarization technique that involves two basic operations. The first operation involves finding coherent chunks in the document and the second operation involves ranking the text in the individual coherent chunks and picking the sentences that rank above a given threshold. The coherent chunks are formed by exploiting the lexical relationship between adjacent sentences in the document. Occurrence of words through repetition or relatedness by sense relation plays a major role in forming a cohesive tie. The proposed text ranking approach is based on a graph theoretic ranking model applied to text summarization task.

1 Introduction

Automated summarization is an important area in NLP research. A variety of automated summarization schemes have been proposed recently. NeATS (Lin and Hovy, 2002) is a sentence position, term frequency, topic signature and term clustering based approach and MEAD (Radev et al., 2004) is a centroid based approach. Iterative graph based Ranking algorithms, such as Kleinberg's HITS algorithm (Kleinberg, 1999) and Google's Page-Rank (Brin and Page, 1998) have been traditionally and successfully used in web-link analysis, social

networks and more recently in text processing applications (Mihalcea and Tarau, 2004), (Mihalcea et al., 2004), (Erkan and Radev, 2004) and (Mihalcea, 2004). These iterative approaches have a high time complexity and are practically slow in dynamic summarization. Proposals are also made for coherence based automated summarization system (Silber and McCoy, 2000).

We propose a novel text summarization technique that involves two basic operations, namely finding coherent chunks in the document and ranking the text in the individual coherent chunks formed.

For finding coherent chunks in the document, we propose a set of rules that identifies the connection between adjacent sentences in the document. The connected sentences that are picked based on the rules form coherent chunks in the document. For text ranking, we propose an automatic and unsupervised graph based ranking algorithm that gives improved results when compared to other ranking algorithms. The formation of coherent chunks greatly improves the amount of information of the text picked for subsequent ranking and hence the quality of text summarization.

The proposed text ranking technique employs a hybrid approach involving two phases; the first phase employs word frequency statistics and the second phase involves a word position and string pattern based weighing algorithm to find the weight of the sentence. A fast running time is achieved by using a compression hash on each sentence.

This paper is organized as follows: section 2 discusses lexical cohesion, section 3 discusses the text ranking algorithm and section 4 describes the summarization by combining lexical cohesion and summarization.

2 Lexical Cohesion

Coherence in linguistics makes the text semantically meaningful. It is achieved through semantic features such as the use of deictic (a deictic is an expression which shows the direction. ex: that, this.), anaphoric (a referent which requires an antecedent in front. ex: he, she, it.), cataphoric (a referent which requires an antecedent at the back.), lexical relation and proper noun repeating elements (Morris and Hirst, 1991). Robert De Beaugrande and Wolfgang U. Dressler define coherence as a “continuity of senses” and “the mutual access and relevance within a configuration of concepts and relations” (Beaugrande and Dressler, 1981). Thus a text gives meaning as a result of union of meaning or senses in the text.

The coherence cues present in a sentence are directly visible when we go through the flow of the document. Our approach aims to achieve this objective with linguistic and heuristic information.

The identification of semantic neighborhood, occurrence of words through repetition or relatedness by sense relation namely *synonyms*, *hyponyms* and *hypernym*, plays a major role in forming a cohesive tie (Miller et al., 1990).

2.1 Rules for finding Coherent chunks

When parsing through a document, the relationship among adjacent sentences is determined by the *continuity* that exists between them.

We define the following set of rules to find coherent chunks in the document.

Rule 1

The presence of *connectives* (such as *accordingly*, *again*, *also*, *besides*) in present sentence indicates the connectedness of the present sentence with the previous sentence. When such *connectives* are found, the adjacent sentences form coherent chunks.

Rule 2

A 3rd person pronominal in a given sentence refers to the antecedent in the previous sentence, in such a way that the given sentence gives the complete meaning with respect to the previous sentence. When such adjacent sentences are found, they form coherent chunks.

Rule 3

The reappearance of NERs in adjacent sentences is an indication of connectedness. When such adjacent sentences are found, they form coherent chunks.

Rule 4

An ontology relationship between words across sentences can be used to find semantically related words across adjacent sentences that appear in the document. The appearance of related words is an indication of its coherence and hence forms coherent chunks.

All the above rules are applied incrementally to achieve the complete set of coherent chunks.

2.1.1 Connecting Word

The ACE Corpus was used for studying the coherence patterns between adjacent sentences of the document. From our analysis, we picked out a set of keywords such that, the appearance of these keywords at the beginning of the sentence provide a strong lexical tie with the previous sentence.

The appearance of the keywords “*accordingly*, *again*, *also*, *besides*, *hence*, *henceforth*, *however*, *incidentally*, *meanwhile*, *moreover*, *namely*, *nevertheless*, *otherwise*, *that is*, *then*, *therefore*, *thus*, *and*, *but*, *or*, *yet*, *so*, *once*, *so that*, *than*, *that*, *till*, *whenever*, *whereas* and *wherever*”, at the beginning of the present sentence was found to be highly coherent with the previous sentence.

Linguistically a sentence cannot start with the above words without any related introduction in the previous sentence.

Furthermore, the appearance of the keywords “*consequently*, *finally*, *furthermore*”, at the beginning or middle of the present sentence was found to be highly cohesive with the previous sentence.

Example 1

- 1. a The train was late.
- 1. b *However* I managed to reach the wedding on time.

In Example 1, the connecting word *however* binds with the situation of the train being late.

Example 2

- 1. a The cab driver was late.
- 1. b The bike tyre was punctured.
- 1. c The train was late.
- 1. d *Finally*, I managed to arrive at the wedding on time by calling a cab.

Example 3

- 1. a The cab driver was late.
- 1. b The bike tyre was punctured.
- 1. c The train was late.
- 1. d I could not wait any more; I *finally* managed to reach the wedding on time by calling a cab.

In Example 2, the connecting word *finally* binds with the situation of him being delayed. Similarly, in Example 3, the connecting word *finally*, though it comes in the middle of the sentence, it still binds with the situation of him being delayed.

2.1.2 Pronominals

In this approach we have a set of pronominals which establishes coherence in the text. From our analysis, it was observed that if the pronominals “*he, she, it, they, her, his, hers, its, their, theirs*”, appear in the present sentence; its antecedent may be in the same or previous sentence.

It is also found that if the pronominal is not possessive (i.e. the antecedent appears in the previous sentence or previous clause), then the present sentence and the previous sentences are connected. However, if the pronominal is possessive then it behaves like reflexives such as “*himself, herself*” which has subject as its antecedent. Hence the possibility of connecting it with the previous sentence is very unlikely. Though pronominal resolution cannot be done at a window size of 2 alone, still we are looking at window size 2 alone to pick guaranteed connected sentences.

Example 4

- 1. a *Ravi* is a good boy.
- 1. b *He* always speaks the truth.

In Example 4, the pronominal *he* in the second sentence refers to the antecedent *Ravi* in the first sentence.

Example 5

- 1. a *He* is the one who got the first prize.

In example 5 the pronominal *he* is possessive and it doesn't need an antecedent to convey the meaning.

2.1.3 NERs Reappearance

Two adjacent sentences are said to be coherent when both the sentences contain one or more reappearing nouns.

Example 6

- 1. a *Ravi* is a good boy.
- 1. b *Ravi* scored good marks in exams.

Example 7

- 1. a The *car* race starts at noon.
- 1. b Any *car* is allowed to participate.

Example 6 and Example 7 demonstrates the coherence between the two sentences through reappearing nouns.

2.1.4 Thesaurus Relationships

WordNet covers most of the sense relationships. To find the semantic neighborhood between adjacent sentences, most of the lexical relationships such as synonyms, hyponyms, hypernyms, meronyms, holonyms and gradation can be used (Fellbaum 1998). Hence, semantically related terms are captured through this process.

Example 8

- 1. a The *bicycle* has two wheels.
- 1. b The *wheels* provide speed and stability.

In Example 8, *bicycle* and *wheels* are related through *bicycle* is the *holonym* of *wheels*.

2.2 Coherence Finding Algorithm

The algorithm is carried out in four phases. Initially, each of the 4 cohesion rules is individually applied over the given document to give coherent chunks. Next, the coherent chunks obtained in each

phases are merged together to give the global coherent chunks in the document.

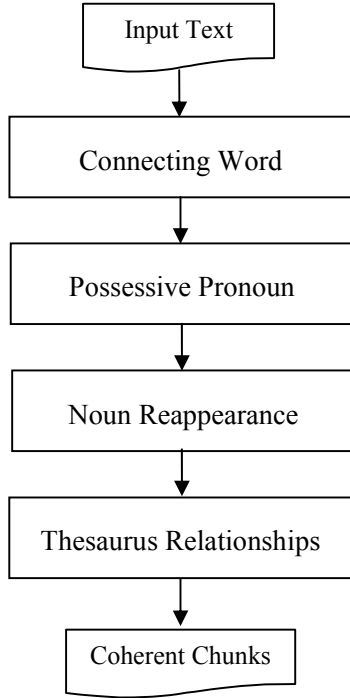


Figure 1: Flow of Coherence chunker

Figure 1, shows the flow and rule positions in the coherence chunk identification module.

2.3 Evaluation

One way to evaluate the coherence finding algorithm is to compare against human judgments made by readers, evaluating against text pre marked by authors and to see the improved result in the computational task. In this paper we will see the computational method to see the improved result.

3 Text Ranking

The proposed graph based text ranking algorithm consists of three steps: (1) Word Frequency Analysis; (2) A word positional and string pattern based weight calculation algorithm; (3) Ranking the sentences by normalizing the results of step (1) and (2).

The algorithm is carried out in two phases. The weight metric obtained at the end of each phase is

averaged to obtain the final weight metric. Sentences are sorted in non ascending order of weight.

3.1 Graph

Let $G(V, E)$ be a weighted undirected complete graph, where V is set of vertices and E is set of weighted edges.

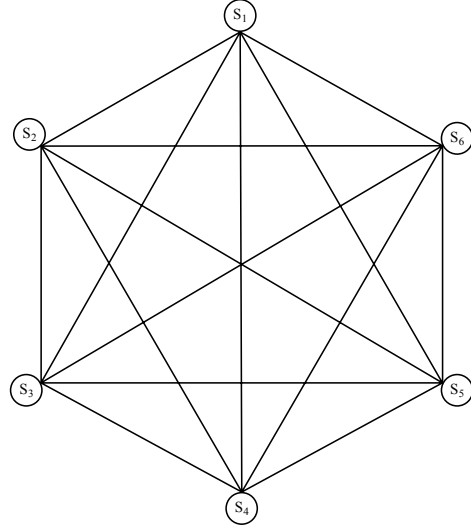


Figure 2: A complete undirected graph

In figure 2, the vertices in graph G represent the set of all sentences in the given document. Each sentence in G is related to every other sentence through the set of weighted edges in the complete graph.

3.2 Phase 1

Let the set of all sentences in document $S = \{s_i \mid 1 \leq i \leq n\}$, where n is the number of sentences in S . The sentence weight (SW) for each sentence is calculated by average affinity weight of words in it. For a sentence $s_i = \{w_j \mid 1 \leq j \leq m_i\}$ where m_i is the number of words in sentence s_i ($1 \leq i \leq n$) the affinity weight AW of a word w_j is calculated as follows:

$$AW(w_j) = \frac{\sum_{\forall w_k \in S} IsEqual(w_j, w_k)}{WC(S)} \quad (1)$$

where S is the set of all sentences in the given document, w_k is a word in S , $WC(S)$ is the total number of words in S and function $IsEqual(x, y)$ returns an integer count 1 if x and y are equal else integer count 0 is returned by the function.

Next, we find the sentence weight $SW(s_i)$ of each sentence s_i ($1 \leq i \leq n$) as follows:

$$SW(s_i) = \frac{1}{m_i} \sum_{w_j \in s_i} AW(w_j) \quad (2)$$

At the end of phase 1, the graph vertices hold the sentence weight as illustrated in figure 4.

[1]"The whole show is dreadful," she cried, coming out of the menagerie of M. Martin.
 [2]She had just been looking at that daring speculator "working with his hyena" to speak in the style of the program.
 [3]"By what means," she continued, "can he have tamed these animals to such a point as to be certain of their affection for."
 [4]"What seems to you a problem," said I, interrupting, "is really quite natural."
 [5]"Oh!" she cried, letting an incredulous smile wander over her lips.
 [6]"You think that beasts are wholly without passions?" Quite the reverse; we can communicate to them all the vices arising in our own state of civilization.

Figure 2: Sample text taken for the ranking process.

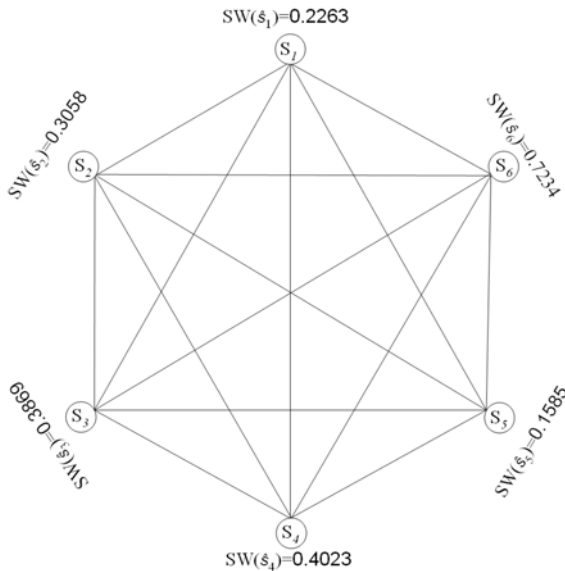


Figure 4: Sample graph of Sentence weight calculation in phase 1

3.3 Compression hash

A fast compression hash function over word w is given as follows:

$$H(w) = (c_1 a^{k-1} + c_2 a^{k-2} + c_3 a^{k-3} + \dots + c_k a^0) \bmod p \quad (3)$$

where $w = \{c_1, c_2, c_3 \dots c_k\}$ is the ordered set of ASCII equivalents of alphabets in w and k the total number of alphabets in w . The choice of $a=2$ permits the exponentiations and term wise multiplications in equation 3 to be binary shift operations on a micro processor, thereby speeding up the hash computation over the text. Any lexicographically ordered bijective map from character to integer may be used to generate set w . The recommendation to use ASCII equivalents is solely for implementation convenience. Set $p = 26$ (for English), to cover the sample space of the set of alphabets under consideration.

Compute $H(w)$ for each word in sentence s_i to obtain the hashed set

$$H(s_i) = \{H(w_1), H(w_2) \dots H(w_m)\} \quad (4)$$

Next, invert each element in the set $H(s_i)$ back to its ASCII equivalent to obtain the set

$$\hat{H}(\hat{s}_i) = \{H(\hat{c}_1), H(\hat{c}_2) \dots H(\hat{c}_m)\} \quad (5)$$

Then, concatenate the elements in set $\hat{H}(s_i)$ to obtain the string \hat{s}_i ; where \hat{s}_i is the compressed representation of sentence s_i . The hash operations are carried out to reduce the computational complexity in phase 2, by compressing the sentences and at the same time retaining their structural properties, specifically word frequency, word position and sentence patterns.

3.4 Levenshtein Distance

Levenshtein distance (LD) between two strings $string1$ and $string2$ is a metric that is used to find the number of operations required to convert $string1$ to $string2$ or vice versa; where the set of possible operations on the character is insertion, deletion, or substitution.

The LD algorithm is illustrated by the following example

LD (ROLL, ROLE) is 1
 LD (SATURDAY, SUNDAY) is 3

3.5 Levenshtein Similarity Weight

Consider two strings, *string1* and *string2* where ls_1 is the length of *string1* and ls_2 be the length of *string2*. Compute $MaxLen = \max(ls_1, ls_2)$. Then LSW between *string1* and *string2* is the difference between $MaxLen$ and LD , divided by $MaxLen$. Clearly, LSW lies in the interval 0 to 1. In case of a perfect match between two words, its LSW is 1 and in case of a total mismatch, its LSW is 0. In all other cases, $0 < LSW < 1$. The LSW metric is illustrated by the following example.

$$\begin{aligned} LSW(ABC, ABC) &= 1 \\ LSW(ABC, XYZ) &= 0 \\ LSW(ABCD, EFD) &= 0.25 \end{aligned}$$

Hence, to find the Levenshtein similarity weight, first find the Levenshtein distance LD using which LSW is calculated by the equation

$$LSW(\hat{s}_i, \hat{s}_j) = \frac{MaxLen(\hat{s}_i, \hat{s}_j) - LD(\hat{s}_i, \hat{s}_j)}{MaxLen(\hat{s}_i, \hat{s}_j)} \quad (6)$$

where, \hat{s}_i and \hat{s}_j are the concatenated string outputs of equation 5.

3.6 Phase 2

Let $S = \{s_i \mid 1 \leq i \leq n\}$ be the set of all sentences in the given document; where n is the number of sentences in S . Further, $s_i = \{w_j \mid 1 \leq j \leq m\}$, where m is the number of words in sentence s_i .

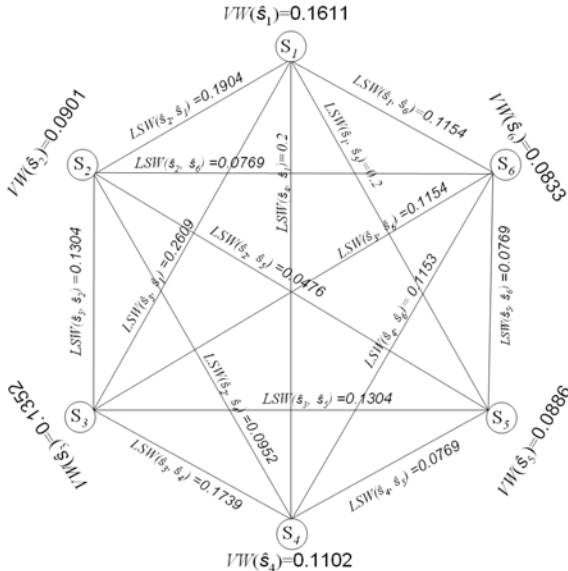


Figure 5: Sample graph for Sentence weight calculation in phase 2

$\forall s_i \in S$, find $\hat{H}(\hat{s}_i) = \{H(\hat{c}_1), H(\hat{c}_2) \dots H(\hat{c}_m)\}$ using equation 3 and 4. Then, concatenate the elements in set $\hat{H}(\hat{s}_i)$ to obtain the string \hat{s}_i ; where \hat{s}_i is the compressed representation of sentence s_i .

Each string \hat{s}_i ; $1 \leq i \leq n$ is represented as the vertex of the complete graph as in figure 5 and $\hat{S} = \{\hat{s}_i \mid 1 \leq i \leq n\}$. For the graph in figure 5, find the Levenshtein similarity weight LSW between every vertex using equation 6. Find vertex weight (VW) for each string \hat{s}_i ; $1 \leq i \leq n$ by

$$VW(\hat{s}_i) = \frac{1}{n} \sum_{\forall \hat{s}_i \neq \hat{s}_j \in \hat{S}} LSW(\hat{s}_i, \hat{s}_j) \quad (7)$$

4 Text Ranking

The rank of sentence s_i ; $1 \leq i \leq n$ is computed as

$$Rank(s_i) = \frac{SW(s_i) + VW(\hat{s}_i)}{2}; 1 \leq i \leq n \quad (8)$$

where, $SW(s_i)$ is calculated by equation 2 of phase 1 and $VW(\hat{s}_i)$ is found using equation 7 of phase 2. Arrange the sentences s_i ; $1 \leq i \leq n$, in non increasing order of their ranks.

$SW(s_i)$ in phase 1 holds the sentence affinity in terms of word frequency and is used to determine the significance of the sentence in the overall ranking scheme. $VW(\hat{s}_i)$ in phase 2 helps in the overall ranking by determining largest common subsequences and other smaller subsequences then assigning weights to it using LSW . Further, since named entities are represented as strings, repeated occurrences are weighed efficiently by LSW , thereby giving it a relevant ranking position.

5 Summarization

Summarization is done by applying text ranking over the global coherent chunks in the document. The sentences whose weight is above the threshold is picked and rearranged in the order in which the sentences appeared in the original document.

6 Evaluation

The ROUGE evaluation toolkit is employed to evaluate the proposed algorithm. ROUGE, an automated summarization evaluation package based on Ngram statistics, is found to be highly correlated with human evaluations (Lin and Hovy, 2003a).

The evaluations are reported in ROUGE-1 metrics, which seeks unigram matches between the generated and the reference summaries. The ROUGE-1 metric is found to have high correlation with human judgments at a 95% confidence level and hence used for evaluation. (Mihalcea and Tarau, 2004) a graph based ranking model with Rouge score 0.4904, (Mihalcea, 2004) Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization with Rouge score 0.5023.

Table 1 shows the ROUGE Score of 567 news articles provided during the Document Understanding Evaluations 2002(DUC, 2002) using the proposed algorithm without the inclusion of coherence chunker module.

	Score
ROUGE-1	0.5103
ROUGE-L	0.4863

Table 1: ROUGE Score for the news article summarization task without coherence chunker, calculated across 567 articles.

Table 2 shows the ROUGE Score of 567 news articles provided during the Document Understanding Evaluations 2002(DUC, 2002) using the proposed algorithm after the inclusion of coherence chunker module.

	Score
ROUGE-1	0.5312
ROUGE-L	0.4978

Table 2: ROUGE Score for the news article summarization task with coherence chunker, calculated across 567 articles.

Comparatively Table 2, which is the the ROUGE score for summary including the coherence chunker module gives better result.

7 Related Work

Text extraction is considered to be the important and foremost process in summarization. Intuitively, a hash based approach to graph based ranking algorithm for text ranking works well on the task of extractive summarization. A notable study report on usefulness and limitations of automatic sentence extraction is reported in (Lin and Hovy, 2003b), which emphasizes the need for efficient algorithms for sentence ranking and summarization.

8 Conclusions

In this paper, we propose a coherence chunker module and a hash based approach to graph based ranking algorithm for text ranking. In specific, we propose a novel approach for graph based text ranking, with improved results comparative to existing ranking algorithms. The architecture of the algorithm helps the ranking process to be done in a time efficient way. This approach succeeds in grabbing the coherent sentences based on the linguistic and heuristic rules; whereas other supervised ranking systems do this process by training the summary collection. This makes the proposed algorithm highly portable to other domains and languages.

References

- ACE Corpus. NIST 2008 Automatic Content Extraction Evaluation(ACE08).
<http://www.itl.nist.gov/iad/mig/tests/ace/2008/>
- Brin and L. Page. 1998. The anatomy of a large scale hypertextualWeb search engine. *Computer Networks and ISDN Systems*, 30 (1 – 7).
- Erkan and D. Radev. 2004. Lexpagerank: Prestige in multi document text summarization. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, July.
- Fellbaum, C., ed. WordNet: An electronic lexical database. *MIT Press, Cambridge* (1998).
- Kleinberg. 1999. Authoritative sources in a hyper-linked environment. *Journal of the ACM*, 46(5):604-632.

- Lin and E.H. Hovy. From Single to Multi document Summarization: A Prototype System and its Evaluation. *In Proceedings of ACL-2002*.
- Lin and E.H. Hovy. 2003a. Automatic evaluation of summaries using n-gram co-occurrence statistics. *In Proceedings of Human Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May.
- Lin and E.H. Hovy. 2003b. The potential and limitations of sentence extraction for summarization. *In Proceedings of the HLT/NAACL Workshop on Automatic Summarization*, Edmonton, Canada, May.
- Mihalcea. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. *In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004) (companion volume)*, Barcelona, Spain.
- Mihalcea and P. Tarau. 2004. TextRank - bringing order into texts. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, Barcelona, Spain.
- Mihalcea, P. Tarau, and E. Figa. 2004. PageRank on semantic networks, with application to word sense disambiguation. *In Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.
- Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K. J. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography* (1990).
- Morris, J., Hirst, G. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* (1991).
- Radev, H. Y. Jing, M. Stys and D. Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40: 919-938, 2004.
- Robert de Beaugrande and Wolfgang Dressler. Introduction to Text Linguistics. *Longman*, 1981.
- Silber, H. G., McCoy, K. F. Efficient text summarization using lexical chains. *In Proceedings of Intelligent User Interfaces*. (2000).

NE Tagging for Urdu based on Bootstrap POS Learning

Smruthi Mukund

Dept. of Computer Science and Engineering
University at Buffalo, SUNY
Amherst, NY, USA
smukund@buffalo.edu

Rohini K. Srihari

Dept. of Computer Science and Engineering
University at Buffalo, SUNY
Amherst, NY, USA
rohini@cedar.buffalo.edu

Abstract

Part of Speech (POS) tagging and Named Entity (NE) tagging have become important components of effective text analysis. In this paper, we propose a bootstrapped model that involves four levels of text processing for Urdu. We show that increasing the training data for POS learning by applying bootstrapping techniques improves NE tagging results. Our model overcomes the limitation imposed by the availability of limited ground truth data required for training a learning model. Both our POS tagging and NE tagging models are based on the Conditional Random Field (CRF) learning approach. To further enhance the performance, grammar rules and lexicon lookups are applied on the final output to correct any spurious tag assignments. We also propose a model for word boundary segmentation where a bigram HMM model is trained for character transitions among all positions in each word. The generated words are further processed using a probabilistic language model. All models use a hybrid approach that combines statistical models with hand crafted grammar rules.

1 Introduction

The work here is motivated by a desire to understand human sentiment and social behavior through analysis of verbal communication. Newspapers reflect the collective sentiments and emotions of the people and in turn the society to which they cater to. Not only do they portray an event that has taken place as is, but they also reveal details about

the intensity of fear, imagination, happiness and other emotions that people express in relation to that event. Newspaper write ups, when analyzed over these factors - emotions, reactions and behavior - can give a broader perspective on the culture, beliefs and the extent to which the people in the region are tolerant towards other religions. Our final goal is to automate this kind of behavioral analysis on newspaper articles for the Urdu language. Annotated corpus that tag six basic human emotions, “happy”, “fear”, “sad”, “surprise”, “anger” and “disgust”, based on the code book developed using the MPQA standards as guideline, is currently being developed. Articles from two leading Urdu newswires, BBC Urdu¹ and Jung Daily² form our corpus.

In order to achieve our goal, it was required to generate the basic tools needed for efficient text analysis. This includes NE tagging and its precursor, POS tagging. However, Urdu, despite being spoken by over 100 million people, (Gordon, 2005) is still a less privileged language when it comes to the availability of resources on the internet. Developing tools for a language with limited resources is a challenge, but necessary, as the volume of Urdu text on the internet is rising. Huda (2001) shows that Urdu has now gained importance on the web, making it the right time to tackle these issues.

It is useful to first examine some basic properties of Urdu and how they affect the cascade of NLP steps in text analysis. Urdu has the *nastaleeq* and *nasq* style of writing that is similar to Arabic

¹ <http://www.bbc.co.uk/urdu/>

² <http://www.jang.net/urdu/>

and flows from right to left (Ahmad et al., 2001). It also adopts some of its vocabulary from Arabic. However, the grammar and semantics of the language is similar to Hindi and this makes it very different from Arabic. For effective text analysis, a thorough syntactic and semantic understanding of the language is required. Detailed grammatical analysis provided by Platts (1909) and Schmidt (1999) can be used for this purpose. The first step in the information retrieval pipeline is tokenization. Unlike English, where the word delimiter is mostly a space, Urdu is more complex. There are space insertion as well as space deletion problems. This makes tokenization a difficult task. The word segmentation model that we propose here combines the statistical approach that considers bigram transition of characters based on their positions in a word and morphological rules with lexicon lookups.

POS tagging comes next in the NLP text analysis pipeline. The accuracy of the tagging model varies, depending on the tagsets used and the domain of the ground truth data. There are two main tagsets designed for Urdu, the CRULP tagset³ and the U1-tagset (Hardie 2003). The U1-tagset, released as a part of EMILLE⁴ corpus, is based on the EAGLES standards (Leech and Wilson 1999). We decided to use the standards proposed by CRULP for the following reasons.

1. The tagset, though not as detailed as the one proposed in U1-tagset, covers all the basic requirements needed to achieve our final goal.
2. The tagged corpus provided by CRULP is newswire material, similar to our final corpus.

A person, when asked to identify an NE tagged word in a sentence would typically try to first find the word associated with a proper noun or a noun, and then assign a suitable NE tag based on the context. A similar approach is used in our model, where the learning happens on the data that is POS tagged as well as NE tagged. Features are learnt from the POS tags as well as the NE tags. The final output of our complete model returns the POS tags

³

http://www.crupl.org/Downloads/ling_resources/parallelcorpus/Urdu POS Tagset.pdf

⁴ <http://www.emille.lancs.ac.uk/>

and NE tags associated with each word. Since we have limited data for training both the POS as well as the NE models, we propose a technique called bootstrapping that helps in maximizing the learning for efficient tagging.

The remainder of the paper is organized as follows. Section 2 discusses the resources assimilated for the work followed by tokenization and word segmentation in Section 3. Section 4 gives a detailed explanation of our model starting with a brief introduction of the learning approach used. Rules used for POS tagging and NE tagging are mentioned in subsections of Section 4. Section 5 presents the results and Section 6 concludes the paper. In each section, wherever relevant, previous work and drawbacks are presented.

2 Resources

Based on the style of writing for Urdu, different encoding standards have been proposed. *Urdu Zabta Takthi* - the national standard code page for Urdu and *Unicode* - international standard for multilingual characters are the two proposed and widely used encoding standards. BBC Urdu and Jung Daily are both encoded with Unicode standards and are good sources of data. The availability of online resources for Urdu is not as extensive as other Asian languages like Chinese and Hindi. However, Hussain (2008) has done a good job in assimilating most of the resources available on the internet. The lexicon provided as a part of the EMILLE (2003) data set for Urdu has about 200,000 words. CRL⁵ has released a lexicon of 8000 words as a part of their Urdu data collection. They also provide an NE tagged data set mostly used for morphological analysis. The lexicon includes POS information as well. CRULP⁶ has also provided a lexicon of 149,466 words that contains places, organizations and names of people. As part of the Urdu morphological analyzer provided by Humayoun (2007), a lexicon of about 4,500 unique words is made available. There are a few Urdu-English dictionaries available online and the first online dictionary, compiled by Siddiqi (2008), provides about 24,000 words with their meanings in English.

Getting all the resources into one single compilation is a challenge. These resources were brought

⁵ http://crl.nmsu.edu/Resources/lang_res/urdu.html

⁶ http://www.crupl.org/software/ling_resources/wordlist.htm

together and suitably compiled into a format that can be easily processed by Semantex (Srihari, 2008), a text extraction platform provided by Janya Inc⁷. Lists of places, organizations and names of famous personalities in Pakistan were also compiled using the Urdu-Wikipedia⁸ and NationalMaster⁹. A list of most common names in Pakistan was composed by retrieving data from the various name databases available on the internet.

The word segmentation model uses the Urdu corpus released by CRULP as the training data. This dataset is well segmented. POS tagging model uses data provided by CRULP and NE tagging model uses data provided by CRL.

3 Word Segmentation and Tokenization

Urdu is a language that has both the space insertion and space deletion problems. The Urdu word segmentation problem as mentioned by Durrani (2007) is triggered by its orthographic rules and confusions about the definition of a word. Durrani summarizes effectively, all the problems associated with Urdu word segmentation. Of all the different techniques explored to achieve this objective, traditional techniques like longest and maximum matching depend mostly on the availability of a lexicon that holds all the morphological forms of a word. Such a lexicon is difficult to obtain. It is shown by Theeramunkong et al., (2001), that for a Thai segmentation system, the efficiency drops considerably (from 97% to 82%) making this approach highly lexicon dependent.

Statistical based techniques have applied probabilistic models to solve the problem of word segmentation. Bigram and trigram models are most commonly employed. Using feature based techniques for POS tagging is also very common. These techniques overcome the limitations of statistical models by considering the context around the word for specific words and collocations. There are other models that generate segments by considering word level collation as well as syllable level collation.

However, for a language like Urdu, a model that is purely statistical will fail to yield good segmentation results. A mixed model that considers the morphological as well as semantic features of the

language facilitates better performance as shown by Durrani (2007) where the word segmentation model uses a lexicon for proper nouns and a statistical model that trains over the n -gram probability of morphemes. Maximum matching technique is used to generate word boundaries of the orthographic words that are formed and these are later verified using the POS information. The segments thus generated are ranked and the best ones are accepted. Statistical models that consider character based, syllable based and word based probabilities have shown to perform reasonably well. The Thai segmentation problem was solved by Pornprasertkul (1994) using the character based approach. In our model, we use a combination of character based statistical approach and grammar rules with lexicon lookups to generate word boundaries.

Urdu segmentation problem can be looked at as an issue of inserting spaces between characters. All letters in Urdu, with a few exceptions, have three forms - initial, medial and final. (We do not consider the detached form for word formation). Words are written by joining the letters together and based on the position of the letter in the word, suitable forms are applied. This property of word formation is the crux of our model. The bigram probability of occurrences of each of these characters, based on their positions, is obtained by training over a properly segmented training set. For unknown characters, unknown character models for all the three position of occurrences are also trained. The probability of word occurrence is noted. Along with this, a lexicon rich enough to hold all possible common words is maintained. However, this lexicon does not contain proper nouns. A new incoming sentence that is not segmented correctly is taken and suitable word boundaries are generated by using a combination of morphological rules, lexicon lookups, bigram word probabilities and bigram HMM character model. The following probabilities are estimated and maximized at character level using the Viterbi algorithm. The following are the calculated probabilities:

- (i) $P(ch_{k(\text{medial})} | ch_{k-1(\text{initial})})$ - is the probability of character k being in medial form given character $k-1$ is in initial form.

⁷ <http://www.janyainc.com/>

⁸ <http://ur.wikipedia.com/wiki/>

⁹ <http://www.nationmaster.com/index.php>

- (ii) $P(ch_{k(final)} | ch_{k-1(initial)})$ - is the probability of character k being in final form given character $k-1$ is in initial form.
- (iii) $P(ch_{k(final)} | ch_{k-1(medial)})$ - is the probability of character k being in final form given character $k-1$ is in medial form.
- (iv) $P(ch_{k(medial)} | ch_{k-1(medial)})$ - is the probability of character k being in medial form given character $k-1$ is in medial form.
- (v) $P(ch_{k(initial)} | ch_{k-1(final)})$ - is the probability of character k being in initial form given character $k-1$ is in final form.

Each word thus formed successfully is then verified for morphological correctness. If the word is not valid morphologically, then the window is moved back over 3 characters and at every step the validity of occurrence of the word is noted. Similarly, the window is moved 3 characters ahead and the validity of the word is verified. All words formed successfully are taken and further processed using a language model that considers the bigram occurrence for each word. The unknown word probability is considered here as well. The word with maximum probability is taken as valid in the given context.

Let $\langle w_1 w_2 w_3 \rangle$ be the word formed by the moving window. Then, the word selected, w_s , is given by

$$(vi) w_s = \max \left\{ \begin{array}{l} P(w_1) | P(w_{prev}) \\ P(w_2) | P(w_{prev}) \\ P(w_3) | P(w_{prev}) \end{array} \right\}$$

where w_{prev} is the previous word.

It is also noted that the number of times a transition happens from a syllable set with consonants to a syllable set with vowels, in a word, is no longer than four in most cases as noted below. This factor is also considered for terminating the Viterbi algorithm for each word.

$Ir | aad | ah$ - three transitions

Some of the morphological rules considered while deciding the word boundaries are given below. Word boundary is formed when

1. The word ends with "ﻮ" - *Nun Gunna*
2. The character transitions over to digits
3. Punctuations marks are encountered ('-' is also included)
4. No two 'ye' - *choti ye* come back to back
5. No characters occur in detached form unless they are initials or abbreviations followed by a period
6. If current character is '*alif*' and the previous character is '*ee*' - *bari ye* then the word boundary occurs after '*alif*'

Some of the drawbacks seen in this model are mainly on account of improper identification of proper nouns. If a proper noun is not well segmented, the error propagates through the sentence and typically the next two or three words fail to get segmented correctly. Also, in Urdu, some words can be written in more than one ways. This mostly depends on the diacritics and ambiguity between *bari* and *choti 'ye'*. The training data as well as the test data were not normalized before training. The model shows a precision of 83%. We realized that the efficiency of this model can be improved if phoneme level transitions were taken into consideration. Training has to be increased over more proper nouns and a lexicon for proper nouns lookup has to be maintained. Diacritics that are typically used for beautification should be removed. Words across the documents need to be normalized to one accepted format to assure uniqueness. This involves considerable amount of work and hence, in order to prevent the propagation of error into the NLP text analysis pipeline, we decided to test our subsequent models using pre-segmented data, independent of our word segmentation model.

4 Learning Approaches

A Conditional Random Field (CRF), is an undirected graphical model used for sequential learning. The tasks of POS tagging and NE tagging are both sequential learning tasks and hence this learning approach is a reasonable choice. What follows is a brief outline about CRF. Interested readers are referred to Lafferty et al., (2001), for more information on CRF.

4.1 Conditional Random Fields (CRF)

A linear chain CRF defines a single log-linear probabilistic distribution over the possible tag sequences y for a sentence x

$$p(y | x) = \frac{1}{Z(x)} \exp \sum_{t=1}^T \sum_{k=1}^K \lambda_k f_k(t, y_t, y_{t-1}, x_t)$$

where $f_k(t, y_t, y_{t-1}, x_t)$ is typically a binary function indicating the presence of feature k , λ_k is the weight of the feature, and $Z(x)$ is a normalization function.

$$Z(x) = \sum_y \exp \sum_{t=1}^T \sum_{k=1}^K \lambda_k f_k(t, y_t, y_{t-1}, x_t)$$

This modeling allows us to define features on states (the POS/NE tags) and edges (pairs of adjacent POS/NE tags) combined with observations (eg. words and POS tags for NE estimation). The weights of the features are determined such that they maximize the conditional log-likelihood of the training data:

$$L(\theta) = \sum_{i=1}^N \log(p_{\theta}(y^{(i)} | x^{(i)})).$$

For the actual implementation, CRF++¹⁰, an open source tool that uses the CRF learning algorithm is used. The L-BFGS algorithm¹¹ is used for optimization.

4.2 NE Tagging using POS information

POS tagging is a precursor for all text analysis tasks. Assigning POS tags to words without any ambiguity depends on contextual information and extracting this information is a challenge. For a language like English, several techniques have been proposed that can be broadly classified into statistical, rule based and hybrid approaches (Ekbal, 2007). The general consensus is that approaches like MEMM and HMM, that work well for Hindi, would work well for Urdu as well, since Urdu is grammatically similar to Hindi (Platts, 1909). However, the linguistic and morphological rules used in the post processing steps differ from Hindi because of Urdu's borrowed vocabulary and

style of writing from Arabic. Also, the requirement for such models to work well is the availability of large training data.

Building NE recognizers for languages like Urdu is difficult as there are no concepts like capitalization of characters. Also, most names of people have specific meanings associated with them and can easily be found in a dictionary with different associated meanings. Various learning approaches have been proposed for this task, HMM based learning approach (Bikel et al., 1999), Maximum Entropy Approach (Borthwick, 1999) and CRF approach (McCallum, 2003) are the most popular. Ashish et al., (2009) show an SVM based approach also works well for such tasks. To overcome the problem of limited data availability, we present a method to increase the amount of training data that is available, by using a technique called bootstrapping.

We do not have a training corpus that is manually tagged for both POS and NE. Our training data consists of two different datasets. The dataset used for POS tagging is provided by CRULP and is tagged using their tagset. The dataset used for NE tagging is provided by CRL as a part of their Urdu resource package. The CRL tagset consists of LOCATION, PERSON, ORGANIZATION, DATE and TIME tags. We use only the first three tags in this work.

Our aim is to achieve effective POS tagging and NE tagging by maximizing the use of the available training data. The CRULP dataset (which we call $dataset_{POS}$) is a corpus of 150,000 words that are only POS tagged and the CRL dataset (which we call $dataset_{NE}$) is a corpus of 50,000 words that are only NE tagged. First, we trained a CRF model on $dataset_{NE}$ that uses only the NE information to perform NE recognition. This one stage model was not effective due to the sparseness of the NE tags in the dataset. The model requires more data while training. The obvious and frequently tried approach (Thamar, 2004) is to use the POS information.

Figure 1 shows a two stage model that uses POS information to perform NE tagging. The first stage POS_A performs POS tagging by using a CRF trained model to assign POS tags to each word in a sentence of $dataset_{NE}$. The second stage NE_A performs NE tagging by using another CRF trained model that uses both the POS information as well

¹⁰ <http://crfpp.sourceforge.net/>

¹¹ <http://www.mcs.anl.gov/index.php>

as the NE information, to perform effective NE tagging.

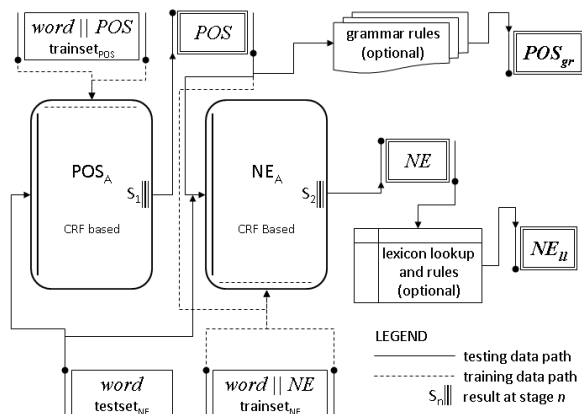


Figure 1. Two stage model for NE tagging using POS information

However, although the accuracy of NE tagging improved over the one stage model, there was scope for further improvement. It is obvious that all the NE tagged words should have the proper noun (NNP) POS tag associated. But, when POS tags were generated for the NE tagged ground truth data in $dataset_{NE}$, most of the words were either tagged as adjectives (JJ) or common nouns (NN). Most tags that come after case markers (CM) were adjectives (JJ) in the training data. Very few accounted for proper nouns after case markers. This adversely affected the NE tagger output. It was also noticed that the POS tagger tagged most of the proper nouns (NNP) as common nouns (NN) because of the sparseness of the proper noun tag in the POS ground truth data set $dataset_{POS}$. This observation made us look to bootstrapping techniques for effective learning.

We propose a four stage model as shown in Figure 2, for NE tagging. Three of the stages are trained using the CRF learning approach and one stage uses a rule based approach. All four stages are trained using unigram features on tags and words and bigram features on tags. The POS tagged dataset, $dataset_{POS}$, consists of words and associated POS tags and the NE tagged dataset, $dataset_{NE}$, consists of words and associated NE tags. We divide both datasets into training and testing partitions. $dataset_{POS}$ is divided into $trainset_{POS}$ and $testset_{POS}$ and $dataset_{NE}$ is divided into $trainset_{NE}$ and $testset_{NE}$.

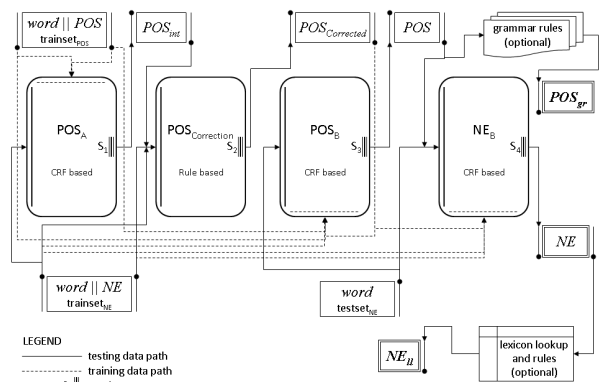


Figure 2. Four stage model for NE tagging using POS information with bootstrapping

In the model shown in Figure 2, POS_A stage is a CRF based stage that is trained using $trainset_{POS}$. Once trained, the POS_A stage takes as input a sentence and generates the associated POS tag for each word in that sentence.

In order to increase the NNP tag associations to improve NE tagging, we generate POS tags for the NE training data in $trainset_{NE}$ using the POS_A stage. The POS tags generated at the POS_A stage are called POS_{int} . The $POS_{correction}$ stage takes as input $trainset_{NE}$ along with its associated POS tags, POS_{int} . At this stage, correction rules - that change the POS tags of NE associated words to proper noun (NNP), assign Case Markers (CM) before and after the NE tags and verify proper tagging of Cardinals (CD) - are applied. The corrected POS tags are called $POS_{corrected}$. A consolidated POS training set consisting of entries from both $trainset_{POS}$ and $trainset_{NE}$ (with $POS_{corrected}$ generated as output from the $POS_{correction}$ stage) is used to train the CRF based POS_B stage. This stage is the final POS tagging stage. Test data consisting of sentences (words) from $testset_{NE}$ is sent as input to stage POS_B and the output generated at stage POS_B is the POS tag associated with each input word of a sentence. The NE_B stage is a CRF based NE tagger that is trained on a dataset consisting of word and associated NE tags from $trainset_{NE}$ and associated POS tags from $POS_{corrected}$. This stage learns from the POS information and the NE information provided in the training data. Once trained, the NE_B stage takes as input words from $testset_{NE}$ and associated POS tags (obtained at stage POS_B) and generates NE tags.

The domain we are interested in is newswire material, and these articles are written in the “jour-

nalistic” or “news writing” style¹². The articles are objective and follow a Subject-Object-Verb structure. Related information is usually presented within close sentence proximity. This makes it possible to hand-craft grammar rules for the discovery of NE tags with fine granularity. The final POS tagged and NE tagged data generated as outputs at stage POS_B and stage NE_B respectively of the four stage model, are processed using rules and lexicon lookups to further improve the overall tagging accuracy of the model. Rules used are mostly domain specific. The rules were applied to the model using Semantex.

4.3 Rules for POS Tagging

1. Our model tags all the Question Words (QW) like ‘کیا’ - *kya* as pronoun (PR). All such occurrences are assigned QW tag.
2. If the word is ‘کیا’ - *kya* and the previous tag is an adjective (JJ) and the next tag is a phrase marker (PM) then assign a light verb tag (VBL) else assign a verb (VB) tag to the word.
3. It was observed that there were spurious instances of proper nouns getting tagged as nouns. In order to correct this error, if a word ends with any of the characters shown below, and the word was tagged as a noun, then the tag on the word was changed to a proper noun.
 ’ی، ’ا، ’ے، ’ین، ’یا، ’بٹ، ’گاہ، ’وٹ، ’گی
4. All valid cardinals were tagged as nouns or proper nouns by the model. This was resolved by looking for a digit in the string.

4.4 Rules for NE Tagging

1. Words like “کورتھ” (court), “بیورو” (bureau), “فوج” (army) etc. are looked up. If there are any nouns or proper nouns above these within a window of two, then the tag on this word is ORGANIZATION.
2. Words like “تنظیم” (organization), “آرمی” are marked ORGANIZATION if the previous word is a proper noun.
3. Lexicon look up for names of places is performed and the POS tag of the next word that is found is checked. If this tag is a

Case Marker (CM) with a feminine gender, like “کے” (main) or “میں”, then the word is marked with a LOCATION tag.

4. If a proper noun that is selected ends with a suffix “pur”, “bad”, “dad” and has the same constraint as mentioned in rule 3, then the LOCATION tag is assigned to it as well.

5 Results

The NE tagging performance, for both the two stage model and the four stage model, are evaluated using Precision (P), Recall (R) and F-Score (FS) metrics, the equations for which are given below.

$$(vii) \quad P = \frac{\text{No. of correctly tagged NEs}}{\text{No. of tagged NEs}}$$

$$(viii) \quad R = \frac{\text{No. of tagged NEs}}{\text{Total no. of NEs in test set}}$$

$$(ix) \quad FS = \frac{2RP}{R + P}$$

We performed a 10 fold cross validation test to determine the performance of the model. The dataset is divided into 10 subsets of approximately equal size. One subset is withheld for testing and the remaining 9 subsets are used for training. This process is repeated for all 10 subsets and an average result is computed. The 10 fold validation test for NE tagging was performed for both the two stage as well as the four stage models.

Set	Two Stage Model			Four Stage Model		
	P	R	FS	P	R	FS
1	48.09	73.25	58.06	60.54	78.7	68.44
2	38.94	72.42	50.65	60.29	80.46	68.93
3	56.98	74.38	64.53	60.54	79.74	68.83
4	38.44	78.05	51.51	60.54	80.79	69.21
5	32.29	75.91	45.31	60.79	80.34	69.21
6	44.82	88.02	59.4	59.31	79.93	68.09
7	45.75	69.75	55.26	61.04	81.73	69.89
8	43.52	71.5	54.11	60.05	80.36	68.74
9	44.64	81.97	57.8	59.93	81.09	68.92
10	44.17	78.18	56.45	60.67	79.22	68.72
Avg	43.764	76.343	55.308	60.37	80.236	68.898

Table 1. NE tagging results for the two stage and four stage models

It can be seen from Table 1 that the four stage model outperforms the two stage model with the

¹² http://en.wikipedia.org/wiki/News_writing

average F-Score being 55.31% for the two stage model and 68.89% for the four stage model.

Table 2 shows the POS tagging results for stages POS_A and POS_B. The POS_B stage performs marginally better than the POS_A stage.

POS _A Results		POS _B Results	
Set	P	Set	P
1	84.38	1	83.97
2	89.32	2	89.84
3	88.09	3	88.48
4	89.45	4	89.66
5	89.66	5	89.76
6	90.57	6	90.63
7	81.1	7	89.24
8	89.47	8	89.5
9	89	9	89.12
10	89.12	10	89.25
Avg	88.016	Avg	88.945

Table 2. POS tagging results for the two stage (POS_A) and four stage (POS_B) models

Although for POS tagging, the improvement is not very significant between the two models, tags like light verbs (VBLL), auxiliary verbs (AUXA and AUXT), adjectives (JJ), demonstratives (DM) and nouns (NN, NNC, NNCM, NNCR) get tagged with higher accuracy in the four stage model as shown in Table 3. This improvement becomes evident in the NE test set. Unfortunately, since this data has no associated POS tagged ground truth, the results cannot be quantified. The *trainset*_{POS} training data had very few instances of proper nouns (NNP) occurring after case markers (CM) and so most of the proper nouns were getting tagged as either adjectives (JJ) or common nouns (NN). After providing more training data to stage POS_B, the model could effectively learn proper nouns. Spurious tagging of adjectives (JJ) and common nouns (NN) reduced while more proper nouns (NNP, NNPC) were tagged accurately and this allowed the NE stage to apply its learning efficiently to the NE test set thereby improving the NE tagging results.

The two stage model tagged 238 NE tagged words as proper nouns out of 403 NE words. The four stage model tagged 340 NE tagged words as proper nouns out of 403 NE words. The four stage model shows an improvement of 25.3% over the two stage model. The results reported for NE and

POS tagging models are without considering rules or lexicon lookups.

POS _A Output		POS _B Output	
Tag	FS	Tag	FS
AUXA	0.801	AUXA	0.816
AUXT	0.872	AUXT	0.898
DM	0.48	DM	0.521
JJ	0.751	JJ	0.765
NN	0.85	NN	0.858
NNC	0.537	NNC	0.549
NNCM	0.909	NNCM	0.923
NNCR	0.496	NNCR	0.51
RB	0.785	RB	0.834
VBLI	0.67	VBLI	0.693
VBT	0.553	VBT	0.586

Table 3. POS tagging results for stages POS_A and POS_B

In order to further improve the POS tagged results and NE tagged results, the rules mentioned in sections 4.3 and 4.4 and lexicon lookups were applied. Table 4 shows the result for NE tagging with an overall F-Score of 74.67%

Tag	NE _A Output		
	P	R	FS
LOCATION	0.78	0.793	0.786
ORGANIZATION	0.775	0.731	0.752
PERSON	0.894	0.595	0.714

Table 4. NE tagging results after applying rules for test results in Table 1

6. Conclusion and Future Work

This work was undertaken as a precursor to achieve our final objective as discussed in Section 1. The basic idea here is to increase the size of the available training data, by using bootstrapping, so as to maximize learning for NE tagging. The proposed four stage model shows an F-Score of 68.9% for NE tagging which is much higher than that obtained by the simple two stage model.

A lot of avenues remain to be explored to further improve the performance of the model. One approach would be to use the bootstrapping technique for NE data as well. However, the rules required can be complicated. More hand crafted rules and detailed lexicon lookups can result in better NE tagging. We have also noticed certain ambiguities in tagging PERSON and LOCATION. Rules that resolve this ambiguity can be explored.

References

- Raymond G. Gordon Jr. (ed.). 2005. *Ethnologue: Languages of the World, Fifteenth edition*. Dallas, TX.: SIL International
- Kashif Huda. 2001. *An Overview of Urdu on the Web*. Annual of Urdu Studies Vol 20.
- Zaheer Ahmad, Jehanzeb Khan Orakzai, Inam Shamsher, Awais Adnan. 2007. *Urdu Nastaleeq Character Recognition*. Proceedings of World Academy of Science, Engineering and Technology. Volume 26, ISSN 2070-3740.
- John T. Platts. 1967. *A grammar of the Hindustani or Urdu language*. Munshiram Manoharlal Delhi.
- R. L. Schmidt. 1999. *Urdu: an essential grammar*. London: Routledge.
- Sarmad Hussain. 2008. *Resources for Urdu Language Processing*. The 6th Workshop on Asian Language Resources.
- P. Baker, A. Hardie, T. McEnery, B.D. Jayaram. 2003. *Corpus Data for South Asian Language Processing*. Proceedings of the 10th Annual Workshop for South Asian Language Processing, EACL.
- M. Humayoun, H. Hammarström, A. Ranta. 2007. *Urdu Morphology, Orthography and Lexicon Extraction*. CAASL-2: The Second Workshop on Computational Approaches to Arabic Script-based Languages, LSA 2007 Linguistic Institute, Stanford University.
- Waseem Siddiqi, Shahab Alam. 2008. Online Urdu-English and English-Urdu dictionary.
- N. Durrani. 2007. *Typology of Word and Automatic Word Segmentation in Urdu Text Corpus*. National University of Computer and Emerging Sciences, Lahore, Pakistan.
- T. Theeramunkong, S. Usanavasin. 2001. *Non-Dictionary Based Thai Word Segmentation Using decision trees*. In proceedings of the First International Conference on Human Language Technology Research, San Diego, California, USA.
- A. Pornprasertkul. 1994. *Thai Syntactic Analysis*. Ph.D Thesis, Asian Institute of Technology.
- Ismat Javed. 1981. قواعد اردو نئی. Taraqqi Urdu Bureau, New Delhi.
- Abdul M. Haq. 1987. نحو و صرف اردو. Amjuman-e-Taraqqi Urdu (Hindi).
- Hassan Sajjad. 2007. *Statistical Part of Speech Tagger for Urdu*. National University of Computer and Emerging Sciences, Lahore, Pakistan.
- John D. Lafferty, Andrew McCallum, Fernando C.N. Pereira. 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282-289.
- John Chen. 2006. *How to use Sequence Tagger*. Semantic Documentation, Janya Inc.
- Bikel, D.M., Schwartz, R.L., Weischedel, R.M. 1999. *An Algorithm that Learns What's in a Name*. Machine Learning 34(1-3), pp. 211-231.
- Borthwick, A. 1999. *Maximum Entropy Approach to Named Entity Recognition*. PhD thesis, New York University.
- McCallum, A., Li, W. 2003. *Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons*. In Proceedings of CoNLL.
- A. Hardie. 2003. *Developing a tagset for automated part-of-speech tagging in Urdu*. Department of Linguistics and Modern English Language, University of Lancaster.
- Leech, G and Wilson, A. 1999. *Standards for tagsets. Edited version of EAGLES Recommendations for the Morphosyntactic Annotation of Corpora*. In van Halteren, H (ed.) Syntactic wordclass tagging. Dordrecht: Kluwer Academic Publishers.
- Awaghad Ashish Krishnarao, Himanshu Gahlot, Amit Srinet and D. S. Kushwaha. 2009. *A Comparative Study of Named Entity Recognition for Hindi Using Sequential Learning Algorithms*. In IEEE International Advance Computing Conference (IACC '09), Thapar University, India. March 6-7.
- Thamar Solario. 2004. *Improvement of Named Entity Tagging by Machine Learning*, Technical Report CCC-04-004, Coordinacin de Ciencias Computacionales.
- Ekbal, A. and Bandyopadhyay, S. 2007. *A Hidden Markov Model Based Named Entity Recognition System: Bengali and Hindi as Case Studies*. Springer LNCS, Vol. 4815, pp. 545.
- R. K. Srihari, W. Li, C. Niu and T. Cornell, "InfoXtract: A Customizable Intermediate Level Information Extraction Engine," *Journal of Natural Language Engineering*, Cambridge U. Press, 14(1), 2008, pp..33-69.

Author Index

Bouma, Gosse, 21

Duarte, Sergio, 21

Filatova, Elena, 30

Hasegawa-Johnson, Mark, 3

Huang, Jui Ting, 3

Islam, Zahurul, 21

K, Sankar, 53

Katragadda, Rahul, 46

L, Sobha, 53

Maganti, Harikrishna, 12

Mukund, Smruthi, 61

Oard, Douglas W., 1

Pingali, Prasad, 46

Raj, Anand Arokia, 12

Savoy, Jacques, 38

Srihari, Rohini K., 61

Varma, Vasudeva, 46

Zhuang, Xiaodan, 3

Zubaryeva, Olena, 38