IIIT Hyderabad in Summarization and Knowledge Base Population at TAC 2011

Vasudeva Varma, Sudheer Kovelamudi, Jayant Gupta, Nikhil Priyatam, Arpit Sood, Harshit Jain, Aditya Mogadala, Srikanth Reddy Vaddepally

International Institute of Information Technology, Hyderabad, India.

Abstract

In this report, we present details about the participation of IIIT Hyderabad in Guided Summarization and Knowledge Base Population tracks at TAC 2011. we have enhanced our summarization system with knowledge based measures. Wikipedia based extraction methods and topic modelling are used to score sentences in guided summarization track. For multilingual summarization task, we investigated the HAL (Hyperspace Analogue to Language Model) where we created a semantic space from word co-occurrences. We show that the results obtained with this unsupervised language independent method are competitive with other state-of-the-art systems. For monolingual and multilingual entity linking task, we extended our previous year's model to a light weight language independent system without utilizing any other external knowledge or resource.

Part I
Guided Summarization Track

1 Introduction

The TAC 2011 Guided Summarization task aims to address two issues simultaneously: Using topics that fall into template-like categories and contain highly predictable elements, as well as explicitly guiding the creation of human reference summaries to contain all these elements. The guided summarization task presents a specific, unified information model that automatic summarizers can emulate. At the same time, emphasis on finding relevant content on the sub-sentential level enables the use of information extraction techniques and other semantic methods, and thus encourages a move towards abstractive summarization. It promotes a deeper linguistic and semantic analysis of source documents. The aim is to generate summaries for a set of newswire articles on a particular topic that is classified into a set of predefined categories. Each category has a list of important aspects and the summary is expected to answer all these aspects while it may also contain other relevant information about the topic.

While information extraction (IE) systems select only specific information to fill the slots of templates, a guided summarization system has to produce a readable summary encompassing all the information about the given templates. Coupling information extraction techniques with summarization is a relatively less explored area, making it hard to find any relevant literature. We combine information extraction, extractive summarization to support user directed multi document summaries. Wikipedia articles are used to build domain knowledge and extract important sentences containing events mentioned in the template. We implemented knowledge based measures, through Wikipedia for extracting concepts from documents and using them instead of simple document words to estimate importance of a sentence.

We investigated the approach of using topic modeling for categorizing the sentences in corresponding

to queries in the template for a given topic.

2 Approach

2.1 Attribute Mining

Words are conventionally considered to be the units of text to calculate importance. Simple word counts and frequencies in the document collection have proved to be working very well in the context of summarization (Varma et al., 2009). This year we extracted attributes of the identified central topic instead of simple word frequencies in computing sentence importance. We built a model that extracts the important attribute words of the central topic in a given set of documents using wikipedia. We have already tested this approach in social media (Kovelamudi et al., 2011). Support vector machines (SVMs), a set of related supervised learning methods for classification and regression analysis are used to facilitate our model. The features on which our system was trained are explained in the following sections.

2.2 Topic Relation using Wikipedia - TR

To understand a context or to identify a topic, we need the words that contain the topic or the set of words that portray the topic. We can reduce this into saying that all we need is the set of related concepts that are been talked about in the context. So, we assume that any topic T can be expressed as $T = \{t_1, t_2, t_3, ...t_n\}$ where 't_i' are the related words discussed in a given context.

The TR feature is all about identifying the list of related key words mentioned in documents that can be found in Wikipedia. Any word that is semantically related to the set of related key words can be included into it, which consequently can take part in conveying the topic. We start with identifying all the words in Wikipedia and then proceed with calculating the most semantically related words among them.

We filter out the words that can be attributes from the key words set using our measures. Any trial to state the semantic relatedness between different words automatically need to draw a huge amount of background knowledge about the concepts that these words represent. One can use hand-crafted lexical structures like thesauri and taxonomies, or statistical analysis of large corpora to process the semantic decisions automatically(Milne, 2007). The limiting factors of such techniques when carried across domains are the background knowledge, precision, scalability and scope. With more than a 18 million articles and thousands of volunteers all over the world, Wikipedia is a growing massive repository of knowledge and is the best alternative when targeted by such limitations.

Wikipedia poses with an immense network of articles, info-boxes, categories, cross-references and explicitly defined semantics which in turn are the marks of its scale and structure.

We only need Wikipedia's structure rather than it's full textual content. We have created SQL database, tables to store and access the page titles and articles fast as suggested by (Milne and Witten, 2009). We map a word in the document set to a Wikipedia article if the word is contained in the Wikipedia article title. We call such words as the $Wikipedia\ words$ and if cannot be mapped, then we call them as $non-Wikipedia\ words$ in the later sections of this paper.

We use the candidate selection method for the primary phase of attribute extraction as stated by (Mihalcea and Csomai, 2007). To process out the most important words from a document using Wikipedia which is given by

$$Importance(a) = Count(N_O)/Count(N_T)$$

Where N_O are the number of Wikipedia articles that have links with word a as anchor text link, where as N_T are the total number of Wikipedia articles that contain it.

To compute semantic relatedness between two words that are mapped to Wikipedia, is equal to finding the semantic relatedness between the articles in Wikipedia to which these words refer. And to do this, the best known way is to compute the relation from the links to those articles in Wikipedia (Medelyan et al., 2008; Milne, 2007).
The relation between two Wikipedia articles x and $y$is given by

$$Relation_{x,y} = 1 - \frac{max(log|A|, log|B|) - log|A \cap B|}{S - min(log|A|, log|B|)}$$

Here A and B are the set of articles which link to

the articles x and y respectively, S is the total number of Wikipedia articles, $A \cap B$ is their overlap. Thus for every $Wikipedia\ word$, we find the semantic relatedness to all the other such words. Topic relation feature (TR) of a word is computed as the sum of its similarity scores with all other such words in the context which is then normalized by the total number of such words. Therefore for a Wikipedia identified words set $\{x_1, x_2, x_3, ...x_k\}$, semantic relatedness of x_i to topic of the context is given by

$$TR_{x_i} = \frac{(\sum_{j=1}^{K} Relation_{x_i,x_j}) - Relation_{x_i,x_i}}{k}$$

The applicability of the above feature is justified in terms of the high scalability and the ever growing knowledge of Wikipedia. The removal of stop words is to avoid computation of semantic similarity for words like 'the' which are also Wikipedia entries.

For the non-Wikipedia words $\{y_1, y_2, y_3, ...y_l\}$ in the document set, the TR feature is modified as the average of all the TR feature values for Wikipedia words, from the same document set. Hence the TR value for any non-Wikipedia word y_i is uniformly given as

$$TR_{y_i} = \frac{\sum_{j=1}^{k} TR_{x_j}}{k}$$

where x_j is a Wikipedia word.

A detailed description of the process of linking entities with Wikipedia and calculating the relatedness measure between two concepts can be found in (Milne and Witten, 2008b) (Milne and Witten, 2008a) The attribute score of a sentence is given by

$$A(s) = \frac{\sum_{w_i \in s} TR(w_i)}{|s|}$$

2.3 Role of prepositions in Estimating Sentence Importance

In English grammar, a preposition is a part of speech that links nouns, pronouns to other phrases in a sentence.

*The red block is **in** the shelf.*
*The head **of** the team lives **in** Delhi.*
*The team is travelling **from** Europe **to** Asia.*

The preposition **'in'** in first sentence is conveying that there is a block, a shelf and some relation between them. Similarly, the other two sentences have some key information about one or more entities and connecting prepositions.

We have used the frequency of a small set of prepositions as a sentence scoring feature. (Varma et al.,) Score of a sentence (s) calculated by PrepImp is given as,

$$PrepImp(s) = \frac{\sum_{w_i \in s} IsPrep(w_i)}{|s|}$$

The list of prepositions used for calculating sentence importance are limited to simple single word prepositions like *in,on,of,at,for,from,to,by,with*, after a careful observation over the data.

2.4 Sentence Ranking

After feature extraction, we estimate a final rank of each sentence using a regression model similar to our earlier Update summarization system in TAC 2009 (Varma et al., 2009). We modeled sentence rank as a dependent variable that is estimated from a set of features. Each sentence in the training phase is represented as a tuple of sentence importance estimate and feature vector. The sentence importance is estimated as the ROUGE 2 score of that sentence with the model summaries. Final rank of the sentence is calculated as,

$$i_s = q(F_s)$$

where i_s is the sentence importance (rank) of sentence s, q is the regression function and F_s is the feature vector comprising all the extracted features.

2.5 Topic Modelling

We have also experimented a topic modelling approach to produce guided summaries. We used Latent Dirichlet Allocation (LDA) where each document may be viewed as a mixture of various topics. We took the help of the package Mallet[1] for implementation of our approach. As the likelihood of a word depends on the other words in the document, we have tagged the sentences in document set with queries in templates of the aspect. We tested our approach for 200 latent topics. The topics which contain the query tagged words were given highest

[1] `http://mallet.cs.umass.edu/topics.php`

importance. The importance factor was raised for all the sentences that contain other words in these query topics. But this constructed model did not lead us much desirable results.

2.6 Summary Extraction

Normally, during summary extraction a subset of ranked list of sentences satisfying redundancy checks, length constraints and other conditions are selected into summary. For one of the runs, among the top ranked sentences as predicted by the regression model we select the sentence having maximum aspect score to the summary. Once an aspect of the template has been answered, we select the next sentence with the maximum score for the remaining aspects in the template. These aspect scores are predicted by LDA trained model that we built.

3 Experiments and Results

The test dataset released by NIST composed of approximately 44 topics, divided into five categories:

- Accidents and Natural Disasters
- Health and Safety
- Attacks
- Endangered Resources
- Investigations and Trials

Each topic has a topic ID, category, title, and 20 relevant documents which have been divided into 2 sets: Document Set A and Document Set B. Each document set has 10 documents, and all the documents in Set A chronologically precede the documents in Set B. No topic narrative is provided; instead the category and its aspects define what information the reader is looking for. Each of the above mentioned categories had a template of aspects that the summary had to answer. For example, the accident category has the following template:

- WHAT: what happened
- WHEN: date, time, other temporal placement markers
- WHERE: physical location
- WHY: reasons for accident
- WHO_AFFECTED: casualties (death, injury), or individuals otherwise negatively affected by the accident
- DAMAGES: damages caused by the accident
- COUNTERMEASURES: countermeasures, rescue efforts, prevention efforts

We used successful features from our earlier work (Varma et al., 2009), Document Frequency Score (DFS), Sentence Location1(SL1) and additionally Prepositional Importance(PrepImp) to produce our first run. The features described in the previous section are used incrementally in this initial experiment. The run had been successfully tested on TAC 2010 guided summarization data which resulted a significant improvement over TAC 2010s best system. The final submission has two runs,

Run1 : DFS, SL1, PrepImp are used as sentence scoring features for cluster A and DFS, SL1, NDR (Varma et al., 2009), PrepImp as features for cluster B. Sentence rank is estimated through regression model from the training feature vectors build over TAC 2009 training data and finally tested on TAC 2010 data.

Run2 : Along with the sentence scoring features used by Run1, KL (Varma et al., 2009), Attribute Extraction with the help of Wikipedia, Topic modeling(LDA) are used as additional features for cluster A. For cluster B, we used DFS, NDR, SL1 are used. The evaluation results of these runs are provided in Table 1

System	ROUGE-SU4	ROUGE-2	Avg-Pyramid Score	Overall Responsiveness
Run1	0.15423 (6/50)	0.11742 (8/50)	0.439	3.045
Run2	0.14790 (12/50)	0.11229 (13/50)	0.403	2.818

Table 1: Evaluation results released by NIST

The results shown in Table 1 are the average scores over all the categories and all the aspects of the template.

Category	Avg-Pyramid Score	Overall Responsiveness
Accidents	0.549	3.111
Attacks	0.502	3.444
Health&Safety	0.343	2.600
Endangered resources	0.326	2.750
Investigations	0.476	3.375

Table 2: Pyramid and Overall Responsiveness scores of Run1 for each category

4 Discussion

Evaluation results show that our Run1 has secured sixth and eighth position in ROUGE-SU4 and Pyramid scores respectively. This time we have utilized our prepositional importance feature in combination with document frequency (DFS) and sentence location(SL). We tested our Run1 on TAC 2010 guided summarization data which yielded better than 2010's best system but unable to take us to the lead on 2011 data. The evaluation results of Run2 are not as good as Run1. Our Wikipedia attribute extraction system which we employed in Run2 intuitively should work for news articles is not outstanding. It implies that we need to tune our attribute extraction more in respect to our current summarization framework in news domain. We believe that the proposed techniques like topic modelling with LDA would work better if the model was built over a sufficiently large news corpus. A closer look at the results reveals that the difficulty of task varies with the type of category. Since we focused on the guided summarization in this submission, it resulted in relatively poor performance in the update task. we strongly believe our new features can be used in a more sophisticated way to devise an effective sentence scoring feature.

Part II
Multilingual Summarization

1 Introduction :

Automated text summarization is a complex and challenging area. A good amount of research has been done in the area of text summarization, where it has been looked from various perspectives categorizing it into different types depending on the way the summaries have been generated, some of them include extractive vs. abstractive, single-document vs. multi-document, language-dependent vs. multilingual (LAST and LITVAK, 2010), etc. Here we are doing Extractive Multilingual summarization where we are extracting the most relevant segments from the text to construct a summary. Of late, the language domain of information in digital form has increased considerably, this is stimulated by the fact that since UTF-8[2] was first introduced in 1993, languages other than English were supported in a computing environment. A further stimulation is provided by the easy availability of Internet which renders easy availability and faster access to large amount of information.

In the realm of such changes all the textual services shall be extended to render support to as many languages as possible. This makes multilingual summarization an interesting problem where newer challenges are arising each day. Text summarization is one such field which requires to be extensible over the language domain. The task of Multilingual Summarization has been introduced for the first time as a pilot task in Text Analysis Conference (TAC - 2011) this year.

1.1 Problem Scenario :

Multi-document summarization is an automatic procedure aimed at extraction of information from multiple texts written about the same topic[3]. Instead of single document summarization, it was a multi document summarization task where we are given a set of news articles which have evolved over a given time. The text documents for summarization were taken from wiki-news which are in English which were translated in all other languages. The corpus for the pilot task consisted of human-created source texts, which were created by native writers per language. Hence, our problem converges to designing of language independent approach to generate a single, fluent, representative summary from a set of documents describing an event sequence.

2 Current approaches :

2.1 Graph-based algorithm for sentence extraction:

The task of sentence extraction is achieved by ranking the complete set of sentences. Each sentence is treated to be a vertex in the graph, where many-to-many weighted edges exist between the vertices. The weights are representing similarity between the two sentences which is a function of content overlap between the two sentences. Content overlap is calculated from common number of tokens between two sentences, this number depends on the filter used to count the common tokens. Length dependency of weights is removed by dividing the content overlap of two sentences by their respective length. Formally, given two sentences S_i ans S_j, with a sentence being represented by the set of N_i words that appear in the sentence: $S_i = W1_i, W2_i, W3_i, ..., WN_i$, the similarity of S_i and S_j is defined as: (Mihalcea, 2005)

$$Similarity(S_i, S_j) = \frac{|W_k|W_k \epsilon S_i \& W_k \epsilon S_j}{log(|S_i|)+log(|S_j|)}$$

This results in a highly connected weighted graph representation of the text. Scores are generated using the existing weighted graph-based ranking algorithms given below.

- Hyperlinked Induced Topic Search (HITS):

$$HITS_A(V_i) = \sum V_j \epsilon In(V_i) HITS_H(V_j)$$

$$HITS_H(V_i) = \sum V_j \epsilon Out(V_i) HITS_A(V_j)$$

- Positional Power Function:

$$POS_P(V_i) = \frac{1}{|V|} \sum_{V_j \epsilon Out(V_i)} (1 + POS_P(V_j))$$

[2] http://en.wikipedia.org/wiki/UTF-8

[3] http://en.wikipedia.org/wiki/Multi-document_summarization

- Page Rank:
$$PR(V_i) = (1-d) + d \times \sum_{V_j \epsilon In(V_i)} \frac{PR(V_j)}{|Out(V_j)|}$$

After the ranking algorithm is run on the graph, sentences are sorted in decreasing order of their score, and the top ranked sentences are selected for inclusion in the summary. The number of sentences selected depends on the requirement of summary.

2.2 Vector based method for sentence extraction :

In this method each sentence is scored based on its content similarity to the rest of the document $(D - S)$. The approach works upon the intuition that the most similar sentence describes the document more appropriately and is more suitable to be included in the summary. $SCORE(S) = sim(S, D - S)$ Where the similarity function (sim) can be any one of the following (LAST and LITVAK, 2010):

- Overlap similarity $\frac{|S \cap T|}{min\{|S|,|D-S|\}}$
- Jaccard similarity $\frac{|S \cap T|}{|S \cup D-S|}$
- Cosine similarity $\cos(\vec{S}, D \vec{-} S) = \frac{\vec{S} \bullet D \vec{-} S}{|\vec{S} \bullet D \vec{-} S|}$

3 Motivation for the approach :

Multilingual summarization requires the summarization approach to be language independent. This shall allow its scalability over multiple languages. This underlying thought motivates for the use of statistical approach, which agrees with our semantic cognitive habits.
In this paper, we investigate the HAL(Hyperspace Analogue to Language) Model where we create a semantic space from word co-occurrences. Since word occurrence marks the scores for each sentence therefore we modeling it into pHAL (Probabilistic HAL) gives us strong base to select sentences. We show that the results obtained with this unsupervised and language independent method is competitive with other state-of-the-art systems.

4 Approach :

The primary algorithm consists of the following steps :

1. Sentence scores are generated using Probabilistic Hyperspace Analogue to Language (Jagadeesh et al., 2005).
2. Dates are extracted for each document.
3. All the sentences(in the complete set of documents) are then sorted in decreasing order of their scores.
4. Top n-sentences are extracted to give a summary. 'n' is given to the system by the user depending upon the size requirement of the summary.
5. These n sentences are then sentences are then sorted increasingly based on the date parameter to provide the temporal sequence to the summary.

5 implementation details :

Generating Scores : From the model of (J et al., 2005) a Hyperspace Analogue to Language model constructs dependancies of a word w on other words based on their occurrence in the context of w in window size K, in a sufficiently large corpus. The pHAL, probabilistic HAL is a natural extension to HAL spaces, as term co-occurence counts can be used to define conditional probabilities. The pHAL can be interpreted as , "given a word w what is the probability of observing another word w' with w in a window of size K".
$pHAL(w'|w) = c \times \frac{HAL(w'|w)}{n(w) \times k}$

The sentence scoring mechanism is based on this model that has been built. Assuming word independence, the relevance of a sentence S can be expressed as,
$P(S|R) = \prod_{w_i \epsilon S} P(w_i|R) \approx \prod_{w_i \epsilon S} P(w_i|S)$

$P(S|R) = \prod_{w_i \epsilon S} \prod P(w_i) P(Q) \prod_{q_j} pHAL(q_j|w_i) \approx \prod_{w_i \epsilon S} P(w_i) \prod_{q_j} pHAL(q_j|w_i)$

5.1 Extracting dates:

All the documents were manually analyzed to identify the pattern for the occurrence of date for each language. The following patterns were most dominant in each of the language.

$$pattern(lang) = \begin{cases} weekday\ date\ month\ year & \text{if } lang \text{ is French} \\ weekday,\ date\ month,\ year\ OR\ weekday,\ month\ date,\ year & \text{if } lang \text{ is Hindi} \\ weekday,\ month\ date,\ year & \text{if } lang \text{ is English} \end{cases}$$

6 Results and Conclusions :

language	Rouge-1	position	Rouge-2	position	Rouge-SU4	position
hindi	0.0685	5	0.02236	2	0.03175	2
english	0.42243	5	0.13985	6	0.17964	5
french	0.402	9	0.12827	9	0.16289	8

Table 3: Official TAC results for the summaries using ROUGE

score	language	position
0.2490644	hindi	4
0.3500278	english	5
0.3583299	french	7

Table 4: Official TAC result for the summaries using AutoSummENG

Based on the intrinsic and extrinsic evaluation performed at TAC are shown in Tables 3 and 4. It can be observed that system is more stable for English given its consistency over various evaluation schemes. For hindi the system performs better than that of the median on the other hand it is equally below median for French .

Part III
Knowledge Base Population (KBP)

1 Introduction

Information overload is one of the biggest challenges faced by web users today.With massive amount of information available on the web it is becoming difficult for the users to search the information they want. In such a scenario rich sources of knowledge like wikipedia are very helpful. The advantage with such Knowledge sources is that they are highly structured thus making the information access very easy. Though knowledge sources like wikipedia offer many advantages even they suffer from few disadvantages: they have to be created and maintained manually therefore not everything in wikipedia is accurate,comprehensive or unbiased. A seemingly obvious solution in such cases is to automate the process of creating or updating Knowledge bases using raw text like news articles. The Knowledge Base Population (KBP) Track at TAC 2011 explores the extraction of information about entities with reference to an external knowledge source. Using basic schema for persons, organizations, and locations, nodes in an ontology are created and populated using unstructured information found in text. A collection of Wikipedia Infoboxes serve as a rudimentary initial knowledge representation. The task is broken down into two subtasks:

- Entity-Linking : In this task given a query a system must be able to point to the Knowledge Base Entry (KB-Entry) to which the query is reffering.If there does not exist a KB-Entry for the query then the system must return NIL. The entity linking system is required to cluster together queries referring to the same non-KB (NIL) entities and provide a unique ID for each cluster, in the form of NILxxxx (e.g., NIL0021). A query consists of two parts: a name string and a background/reference document id.We are provided with two resources: A partially filled Knowledge Base and a corpus containing raw text. The queries can be either mono-lingual(only in English) or cross-lingual(both english and chinese in our case).

- Slot-Filling : Slot filling involves mining facts about an entity and filling in the appropriate values. A slot fillig query consists of a KB-Entry id and a Slot id.The system is supposed to fill in the appropriate slot value or return NIL if that particular slot information is not available in the corpus. We have participated in Mono-Lingual and Cross Lingual Entity-Linking task.

2 PREVIOUS APPROACHES AND MOTIVATION

Most of the previous approaches that solve this problem use heavy amount of world knowledge (External knowledge). For instance many approaches exploit the structure of wikipedia like infoboxes,outlinks,references etc to update the knowledge base (Varma et al.,). Also most of them use heavy resources like part-of-speech taggers and named entity recognizers which are mostly language dependent. This was our main motivation: to build a light weight language independent system which achieves the goal without requiring any external knowledge or resource.

3 PREPROCESSING

- Indexing Knowledge Base (KB) and Document Set: There are about one million documents in the document set and each KB-Entry consist of different attributes like wikititle,name.type,wiki-text etc. Hence for faster retrieval we have indexed both: The document set as well as the Knowledge Base. We have used Indri for indexing purposes because it provides lot of flexibility while indexing and has a very powerful query language.

- Extracted equivalent entities from Knowledge Base: An entity can be reffered in many ways, for instance the entity "Mahatma Gandhi" can be referred as "gandhiji" or "M.K.Gandhi". We have extracted all possible entity synonyms for each entity from the Knowlege base. We have done this using wikititle,Name and external links. The following example would make it more clear. Consider the two KB-Entries.

```
<wikititle>Kasturba Gandhi</wikititle>
<type>PER</type>
<id>E000001</id>
<fact name="Name">Kasturba Gandhi</fact>
<fact name="Wife of">
<link entity_id="E000002">
Mohandas Karamchand Gandhi</link>
</fact>
...
<wiki_text>...
...</wiki_text>
<wikititle> Gandhiji </wikititle>
<type>PER</type>
<id>E000002</id>
<fact name="Name">M.K. Gandhi</fact>
...
```

<wiki_text >...
... </ wiki_text>

From the above two KB-Entries we can infer that the three words "Gandhiji","M.K.Gandhi" and "Monhandas Karamchand Gandhi" are synonymous i.e refer to the same person.

4 APPROACH

Given a query (the name string) we find the list of all possible KB-Entry id's. We call this set as the candidate solution set. Then based on the similarity between the KB-Entry and the reference document we give a similarity score to each candidate solution. We rank the candidate solutions according to the given similarity score and output the KB-Entry id which has the highest score. If the candidate solution set is empty then we return NIL as the solution and consider the query for clustering, this means that there does not exist a KB-Entry similar to the query. The approach can logically be broken down into three steps:

- Finding nearest variations of a query: Even after finding all entity synonyms from the Knowledge base there might be a possibility of missing out on some of the entity/query variations. For this purpose we consider all entities within a particular distance to the query. The distance metric used for this purpose is the Levenshtein distance and the amount of variation allowed is 30 percent. Therefre all the KB-Entities within 30 percent distance of the query (and its synonyms) are considered as the candidate solutions.
- Ranking candidate solutions: The reference document of the query is matched against the "wiki_text" of each entity in the candidate solution set and a similarity score is given to each candidate solution.The similarity metric used here is the cosine similarity. The candidate solution with the highest similarity score is considered as the answer.
- Clustering: For queries whose candidate solution set is empty we consider them for clustering. Two queries can be grouped into one cluster if there is some similarity between their respective name strings (measured using Levenshteins distance) and reference documents (measured using cosine similarity). Each cluster is given an id of the form NILxxxx. Note for cross-lingual queries we have used google translate to convert the queries from chinese to english

5 RESULTS AND FUTURE WORK

The following are the results for monolingual and cross-lingual entity linking systems.

One possible future extension might be to include different facts about an entity (like name,type) while calculating the similarity between the reference document and the KB-Entry. In the present approach we only use the wiki_text of the KB-Entry, ignoring rest of the facts about the entity. As mentioned in the paper,our approach does not make use of any external knowledge or resource, but if we use tools like named entity recognizer and run it on wiki_text and other attributes of an entity we will definitely have more information in our hand.

6 CONCLUSION

In our paper we have provided an approach to monolingual and cross-lingual entity linking. Given a query our system is able to identify the appropriate Knowledge base entry related to it (if such an entry exists), otherwise it returns NIL. For each query (the name string) we find out all query variations using the pre-calculated entity-synonym list and levenshtein distance. We compare the wiki_text attribute of each entity with the reference document and based on the similarity we find the relevance of the query and the entity. The task of knowledge base population is heavily resource dependent so after a certain point we need to use external knowledge and natural language processing tools to attain high accuracies.

References

J. Jagadeesh, P. Pingali, and V. Varma. 2005. A relevance-based language modeling approach to duc 2005. *Proceedings of Document Understanding Conferences (along with HLT-EMNLP 2005), Vancouver, Canada*.

Sudheer Kovelamudi, Sethu Ramalingam, Arpit Sood, and Vasudeva Varma. 2011. Domain independent model for product attribute extraction from user reviews using wikipedia. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1408–1412, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

M. LAST and M. LITVAK. 2010. Language-independent techniques for automated text summarization.

O. Medelyan, I.H. Witten, and D. Milne. 2008. Topic indexing with Wikipedia. In *AAAI WikiAI workshop*.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. CIKM '07, pages 233–242, New York, NY, USA. ACM.

R. Mihalcea. 2005. Language independent extractive summarization. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 49–52. Association for Computational Linguistics.

KBP2010 micro-average	Precision	Recall	F1-scores	type
0.348	0.244	0.274	0.258	Monolingual
0.413	0.372	0.401	0.386	Cross-lingual

David Milne and Ian H Witten. 2008a. An effective, low-cost measure of semantic relatedness obtained from wikipedia links.

David Milne and Ian H. Witten. 2008b. Learning to link with wikipedia. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 509–518, New York, NY, USA. ACM.

D. Milne and I.H. Witten. 2009. An open-source toolkit for mining Wikipedia. In *Proc. New Zealand Computer Science Research Student Conf., NZCSRSC*, volume 9.

D. Milne. 2007. Computing semantic relatedness using wikipedia link structure. In *New Zealand Computer Science Research Student Conference*. Citeseer.

V. Varma, P. Bysani, K. Reddy, V.B. Reddy, S. Kovelamudi, S.R. Vaddepally, R. Nanduri, K. Kumar, S. Gsk, and P. Pingali. Iiit hyderabad in guided summarization and knowledge base population.

Vasudeva Varma, Praveen Bysani, Kranthi Reddy, Vijay Bharat, Santosh GSK, Karuna Kumar, Sudheer Kovelamudi, Kiran Kumar N, and Nitin Maganti. 2009. iiit hyderabad at tac 2009. Technical report, Gaithersburg, Maryland USA.