Event Recognition from News Webpages through Latent Ingredients Extraction

Rui Yan[1], Yu Li[2], Yan Zhang[1,⋆], and Xiaoming Li[1]

[1] School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, P.R. China
[2] School of Computer Science, Beihang University, Beijing 100083, P.R. China
{r-yan,lxm}@pku.edu.cn, carp84@gmail.com, zhy@cis.pku.edu.cn

Abstract. We investigate the novel problem of event recognition from news webpages. "Events" are basic text units containing news elements. We observe that a news article is always constituted by more than one event, namely Latent Ingredients (LIs) which form the whole document. Event recognition aims to mine these Latent Ingredients out. Researchers have tackled related problems before, such as discourse analysis and text segmentation, with different goals and methods. The challenge is to detect event boundaries from plain contexts accurately and the boundary decision is affected by multiple features. Event recognition can be beneficial for topic detection with finer granularity and better accuracy. In this paper, we present two novel event recognition models based on LIs extraction and exploit a set of useful features consisting of context similarity, distance restriction, entity influence from thesaurus and temporal proximity. We conduct thorough experiments with two real datasets and the promising results indicate the effectiveness of these approaches.

Keywords: Event Recognition, Latent Ingredient, Segmentation.

1 Introduction

News webpages increasingly become an essential component of web contents nowadays, and as a result, news flood surges in the Internet. Within the domain of modern information retrieval, news search plays an important role. Contemporary news search is based on document-level retrieval. However, from our observation news documents are not indivisible: they always contain more than one event. An event is defined as "something that happens at a specific time and location"[1]. Events within the same news document are related but to some extent independent from each other. Therefore not all of them are relevant to issued queries. Search engines can instead return fine-grained event-level results to facilitate more accurate search from news webpages and "query-event" match will be more successful than full document in news retrieval. Furthermore, event recognition techniques can stimulate other relevant researches due to its potential use for Topic Detection and Tracking (TDT). The fine granularity of event representation motivates more accurate task results.

⋆ Corresponding author.

P.-J. Cheng et al. (Eds.): AIRS 2010, LNCS 6458, pp. 490–501, 2010.

We illustrate a news report from Xinhua News[1] in Fig.1. The article can be divided into several events and we zoom in three of them: new death caused by Swine Flu in Singapore; retrospection of the first infection in Singapore; a confirmed patient in Malaysia. These events are related but independent. By appropriate event recognition, one can find the most relevant event description with less jeopardized noises. If the news report is compared as a "dish" then these constituent events are ingredients to form the dish, but they are latent and need to be mined. Therefore we name them as "Latent Ingredients" (LIs). LIs are atomic for event-level retrieval.

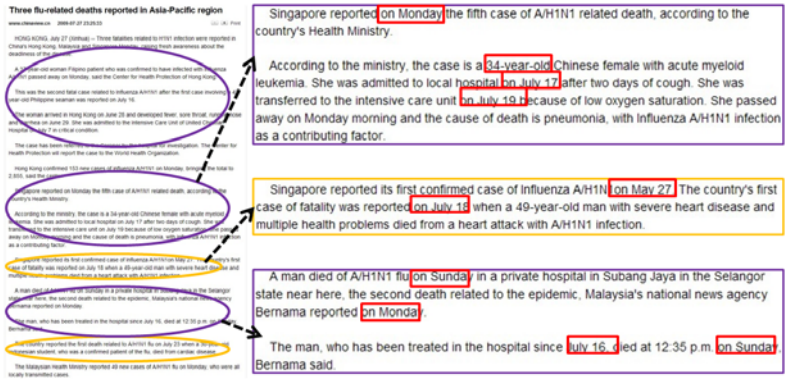

Fig. 1. A news illustration from *Xinhua.net* which consists several Latent Ingredients. Red boxes emphasize temporal information and we tag and use them later in this paper.

The first challenge for event recognition is to distinguish "events" from plain texts. We look into discourse structure and event representation to locate which parts of texts contain events. A more important challenge is to precisely detect event boundaries after we recognize the potential area of events. This is quite different from existed segmentation techniques, e.g. majorly dependent on inter-sentence similarity measurement but ours is event-oriented. Since there are multiple features to affect the procedure, a balance among all features present special difficulties. We manage to decide whether an event shifts or continues with appropriate solutions. We provide two models to address the challenges.

The rest of this paper is organized as follows: in Section 2 we revisit related work. In section 3 we modify the classic TextTiling algorithm into Temporal Textiling Model (TTM). We describe our innovative LIs Growth Model (LGM) based on sentence feature analysis in Section 4. Section 5 presents rich experiments and corresponding results. We draw conclusions in Section 6.

[1] http://www.xinhuanet.com/english/sf/

2 Related Work

2.1 Discourse Structure Analysis

We extract atomic events as LIs, similar to key paragraph extraction [6,7]. Discourse analysis in journalism deals with similar problems. Ponte and Croft used a Gaussian Length Model to weight potential segment length with the prior probability defined in [15]:

$$\frac{k}{\sigma\sqrt{2\pi}} exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{1}$$

where μ is the estimated mean length, σ is the estimated standard deviation and k is a constant for scaling purpose. Grimes proposed a segmentation standard based on time, space, character and theme [8]. In news this standard can be mapped to temporal expressions, entities (location and person) and semantic contexts. Bestgen et al indicated temporal information was used to signal thematic shift in discourse structures [4,5]. We will use these conclusions as the basic assumptions in this work.

2.2 Segmentation Techniques

Text segmentation techniques have gained emphasis through all these years and kept on progressing. Salton discussed the decomposition of text into segments and themes where a segment is a contiguous block of text discussing a single subtopic [16]. Hearst discussed a method named "TextTiling" to segment expository texts into multi-paragraph subtopics which they call "tiles" [12,10]. The text is initially broken up into blocks of size N and then a similarity curve of adjacent blocks is computed using cosine similarity to identify topic boundaries by calculating relative similarity difference. A great variety of research works [11,9,2,18] furthered deeper on classic TextTiling. Hidden Markov Models (HMMs)[17,13] approaches broke texts using a sequential Markov stochastic decision process which generates text fragments relevant to a particular query. In recent years as topic model proved its importance and researchers connected segmentation techniques with topic analysis [14,3].

Our approaches are different from previous ones. Firstly we cannot use a fixed block size because in our datasets, document length varies significantly due to different representation of news. Yet the comparison between neighboring blocks in fixed size is not enough. Besides, our approach is independent of specific queries, unlike HMMs. Finally, our approaches are more event-centered than topic models. We consider more news elements with the help of lexical thesaurus and deal with the problem of few key terms in common between sentences.

3 Temporal TextTiling Model

Based on Bestgen's conclusion [4] and according to our statistics of human annotated data, 87.23% LIs start from a sentence with temporal information. As

illustrated in Fig.1, temporal expressions play a vital part in indicating LIs extraction. We treat the sentence with a timestamp as a head sentence and the LIs extraction starts from it.

3.1 Timestamp Extraction

To identify the head sentences we need to locate temporal expressions. There are specific and non-specific temporal expressions. Specific temporal expressions are meaningful in that they satisfy news elements criterion and indicate events while non-specific ones do not. Specific temporal expressions can be classified as explicit ones, which are simple to recognize, and implicit ones which need semantic inference from reference time point by calculating elapse. Expressions such as "tomorrow" indicate time offsets. Secondly time value can be time points or time intervals. We assign publish time to the whole news article and make references when encounter new time tags during sequential processing within each LI.

Table 1. Categorization of temporal expressions

Meaning	Categorization	Examples
Specific	Explicit expressions	on May 28th
		between 5.9 and 5.11
	Implicit expressions	from Monday to Friday
		after two days
Non-Specific	Useless temporal expressions	34-year-old Chinese
		progress by days of study

We implement a time tagger based on GATE[2] to recognize temporal expressions. The tagger extracts them, discards non-specific ones, makes semantic inference and regulates with uniform format (mm/dd/yyyy). In this work, we use temporal expressions to denote those specific ones.

3.2 Temporal TextTiling

We modified the classic TextTiling algorithm, which uses inter-sentence similarity. Previous TextTiling specifies a fixed size of block as the unit of comparison, and adjacent blocks are compared. However, this measure cannot be directly applied to our scenario due to the character of news representation as mentioned in Section 2. Therefore we regard a sentence as a block.

After extracting and regulating temporal expressions, we locate the first head sentence (s_h) to be the beginning of an LI. All pairs of adjacent sentences from s_h are assigned a similarity value and these values are examined for peaks and valleys. Peak values imply two sentences cohere well whereas valleys indicate potential boundaries. We choose the first boundary from s_h as the end of this

[2] http://gate.ac.uk

LI and move on to the next head sentence with temporal expressions. Given two sentences s_1 and s_2, similarity between sentences is calculated by a cosine measurement, where $w(t, s)$ is the weight of term t in sentence s.

$$similarity = \frac{\sum_{t \in s_1 \cap s_2} w(t, s_1) w(t, s_2)}{\sqrt{\sum_{t \in s_1} w(t, s_1)^2} \times \sqrt{\sum_{t \in s_2} w(t, s_2)^2}} \tag{2}$$

Term Weighting. $w(t, s)$ is measured by $tf.idf$ from standard information retrieval in previous TextTiling. In [9] a twist term weighting is introduced: $w(t, s)$ is the tf value (term frequency) of t within the block, and here the block means the sentence. However, only term frequency cannot ensure the terms in common are rare in other parts of the document but the global $tf.idf$ mechanism based on the whole collection would make the vector too sparse. Thus $tf.isf$ (term frequency-inverse sentence frequency) weighting strategy is utilized. Still there may be a problem. tf within a sentence is often too small and successive multiplication of $tf.isf$ weights may cause an underflow. So we implement a novel term weighting strategy $dtf.isf$. dtf is measured within the local document D, and isf in D indicates whether t is a distinguishing term or not. The four term weighting strategies are compared in our experiments to evaluate their performances in the task of LIs extraction.

4 LIs Growth Model

As specified above, we start LIs extraction from a head sentence s_h with temporal expressions, the indicators of topic drifting. Neighboring contexts tend to describe the same event due to the semantic consecutiveness of natural languages: human discourse is consistent. An LI expands by absorbing sentences that stick to the event. We assume there are two factors deciding relevance to the event. Naturally one is the context similarity between the pending sentence (s_p) and every sentence from the expanding LI (s_L). The other factor is the probability for s_L to belong to LI. Here we denote significance to be the probability of being related to LI. Intuitively, if s_p is similar to a more significant sentence, it is more likely to be relevant to LI. These two factors can be formulated in Equation 3 as follows, where $|LI|$ denotes the number of sentences in the expanding LI:

$$relevanceScore = \frac{\sum_{s_p \cup LI} significance \times similarity}{|LI| + 1} \tag{3}$$

4.1 Context Similarity

Similarity is always an important measurement in text processing. We employ pre-process techniques for accuracy, including Part-of-Speech tagging, stemming,

stop word removal and named entity recognition. Like the Temporal TextTiling Model, similarity values are measured on sentence level using Equation 2.

We will next move on to the sentence feature analysis according to the news principles. Each of these lexical elements is essential for LIs extraction. We present the symbols that would be mentioned later in Table 2.

Table 2. Symbols used in the following sections

Symbols	Meanings
s_p	The pending sentence to decide whether to add
s_h	The head sentence (with temporal expressions)
$\|s\|$	The distance (offset) from s_h
e_L	Named entities contained in LI
e_W	Named entities connected by WordNet
e_i	Named entities contained in sentence i
s_e	A sentence in the LI containing relevant entity (entities)
s_t	A sentence in the LI containing temporal expression(s)

4.2 Distance Restriction

According to the Gaussian Length Model, the tendency to agglomerate attenuates as distance becomes larger from head sentence s_h ($|s_h|$=0). The length of LI *Len* follows Gaussian distribution. $P(X \sim O)$ represents the probability for s_p to contain a common event with the expanding LI, which is a decay caused by distance, namely distance restriction $f_d(x)$ ($x = |s_p|$).

$$f_d(x) = P(X \sim O) = P(X < Len) = \int_X^{+\infty} \frac{k}{\sigma\sqrt{2\pi}} exp(-\frac{(t-\mu)^2}{2\sigma^2})dt \qquad (4)$$

$f_d(x)$ is illustrated in Fig.2. However, Equation 4 based on statistical distance is incomplete. Considering Grimes' theory, besides theme (semantic context) similarity, space and character (i.e. named entities) and time (i.e. temporal expressions) have influences and should all be taken into consideration. We treat these standards homogeneously and they share similar decay function.

4.3 Named Entity Influence

Named entities, such as person, location or organization names, are usually utilized in text mining problems. They are different from plain terms. Relevant named entities in two sentences indicate a probable common event. Therefore we need to identify such entities, which are either entities from current LI (denoted as e_L) or entities connected to e_L through a path by WordNet(e_W). The structure of WordNet is illustrated in Fig.3. Distance from entity A to entity E is 5 with a path $\{A, B, root, C, D, E\}$. Relevant entities form a dynamically growing set during LI expansion. e_L is initiated from e_{s_h} and iteratively updated by adding e_{s_p} when s_p is added to LI:

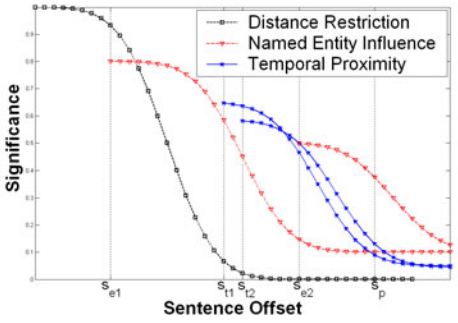

Fig. 2. Decay functions illustration

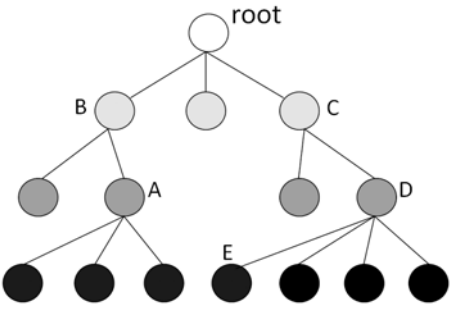

Fig. 3. Hierarchical structure of WordNet

A sentence containing relevant entity(ies) will raise the probability to share the event with LI despite the distance restriction. We use $f_e(x)$ to define the named entity influence and it is illustrated in Fig.2 too.

$$f_e(x) = e^{-\alpha \times dist} \times f_d(x - |s_e|) \tag{5}$$

$dist$ is the distance between e_{s_p} and e_L, which is measured by the length of the shortest path in WordNet. If entities are exactly matched, $dist$ will be 0, then $f_e(x)$ reduces into $f_d(x - |s_e|)$ and the significance of s_e is equal to s_h. Generally there are various entities in e_L, and $dist$ is the average distance $\overline{dist}$ of all entity pairs. α is a scaling factor between 0 and 1.

4.4 Temporal Proximity

We consider temporal proximity contribute to common events as well. Re-mention of adjacent temporal information might strengthen event continuousness, but if two timestamps are too far away, a new event might begin here. When new temporal expressions are identified, we need to decide whether a new LI starts or the original LI still expands. We introduce time decay to measure the temporal proximity in Equation 6 and it is also illustrated in Fig.2:

$$f_t(x) = e^{-\beta \times \frac{\Delta t}{T_d}} \times f_d(x - |s_t|) \tag{6}$$

Δt means the temporal distance between the new time value t_n and the timestamp t_0 in s_h. When refers to expressions designating time intervals, Δt means the closest time distance between the two. T_d is the time span of the news document. β is a also scaling factor between 0 and 1.

We use a significance to measure the probability of being the same event and all features mentioned above impact the significance score. Note that in the expanding LI, there may be more than one s_e or s_t and hence there may be more than one $f_e(x)$ or $f_t(x)$. For the completeness of an LI, we choose the maximum $f_e(x)$ and $f_t(x)$. Equation 7 takes the arithmetic average score of all significance.

$$significance = (f_d(x) + max\{f_e(x)\} + max\{f_t(x)\})/3 \tag{7}$$

We combine both similarity and significance in Equation 3 and obtain a relevance score from all sentence pairs between s_p and s_L. We add s_p into LI when relevance score exceeds a threshold. This LIs extraction model is like a growing snow ball instead of the parallel process of Temporal TextTiling Model (TTM). Therefore we name it as LIs Growth Model (LGM).

5 Experiments and Evaluation

5.1 Data Description

Since there is no existing standard test set for LIs extraction, we opt to construct our own test sets which consist news datasets and golden standards. We construct two separate datasets manually for LIs extraction. One is based on news documents from Automatic Content Extraction 2004 corpus ($ACE04$). Considering these news reports are years away from now, we collect recent news pages from Xinhua news website ($Xinhua$). We sample 1000 news documents from the datasets according to their length distribution, 500 from each, all in English.

5.2 Evaluation Metrics

In performance evaluation we use the *Precision* and *Recall* criterion in information retrieval. Firstly we consider the sentence level performance among LIs. Suppose L_1 is an LI given by human with m sentences and L_2 by computer with n sentences. We treat L_1 and L_2 as a pair when L_1 contains the head sentence s_h of L_2. Sentence level precision (p_{sent}) within is calculated by checking how many sentences in L_2 are found in L_1 and the recall (r_{sent}) is to check how many sentences in L_1 are retrieved. We use formalized evaluation metrics as follows:

$$p_{sent} = \frac{|L_1 \cap L_2|}{n}; r_{sent} = \frac{|L_1 \cap L_2|}{m}; F_{sent} = \frac{2 \times p_{sent} \times r_{sent}}{p_{sent} + r_{sent}} \tag{8}$$

Moreover, we also consider the event level performance to measure how many LIs are missed or falsely alarmed. Suppose there are a LIs in document D found by man, b LIs in D by computer and $|L_1| \cap |L_2|$ is the number of matched LI pairs. Event level precision (p_{event}) indicates how many LIs are correctly found and the recall (r_{event}) is to measure how many events are found by the algorithms.

$$p_{event} = \frac{|L_1| \cap |L_2|}{b}; r_{event} = \frac{|L_1| \cap |L_2|}{a}; F_{event} = \frac{2 \times p_{event} \times r_{event}}{p_{event} + r_{event}} \tag{9}$$

The final F-score in a document D is calculated by the harmonic mean of $\overline{F_{sent}}$ from all LIs within this document and F_{event}.

$$F = \frac{2 \times \overline{F_{sent}} \times F_{event}}{\overline{F_{sent}} + F_{event}} \tag{10}$$

5.3 Parameter Tuning

There are several free parameters in our LIs extraction models. μ, σ and k are fitted to Formula 1 by the LI length statistics. From our statistics of LIs length from 1000 reports, we choose $k = 15$, $\mu = 7$, $\sigma = 2$ in Formula 1.

Next we examine the influence of scaling factor α and β under a specific threshold. During every "training and testing" process, we vary α from 0 to 1 with the step of 0.1 and make the same move to β. We check the F-score when these two parameters change in Fig.4 and get a best α and β value pair (α_{best} and β_{best}) under each given threshold. In *Xinhua* we can see that when β exceeds a certain value the F-score improves significantly (β=0.5 in Fig.4) under all α but α shows little and unstable influence of the overall performance. The best F-score in Fig.4 is achieved when α=0.3 and β= 0.6. In *ACE*04 we observe the best F-score is achieved when α=0.6 and β= 0.5 and the performance varies slightly when α and β are restricted in a particular region. In general the effect of α is weaker than that of β despite of different datasets, but β is more sensitive to datasets than α is.

To the decision of threshold value, we notice *relevanceScore* sometimes varies significantly from document to document. Therefore, we measure the threshold more locally, within each document. In every news article, we locate all potential head sentences and check the *relevanceScore* of each sentence following. Hence we obtain all *relevanceScores* with a range in $[\underline{relScore}, \overline{relScore}]$ and the threshold is computed by $(\underline{relScore} + \gamma \times (\overline{relScore} - \underline{relScore}))$ where γ varies from 0 to 1 at a step of 0.1. Take LGM of Equation 7 as an example, the parameter tuning procedure is listed in Table 3.

5.4 Performance and Discussion

We examine the performance of temporal information extraction. We randomly sample 100 reports and generate timestamps by humans. The time tagger extracted 483 temporal expressions, 441 correctly inferred. The accuracy is 91.30%.

We compare the performance of LIs extraction models. We choose TextTiling as *Baseline*-1, and Temporal TextTiling Model (TTM) as *Baseline*-2. Four

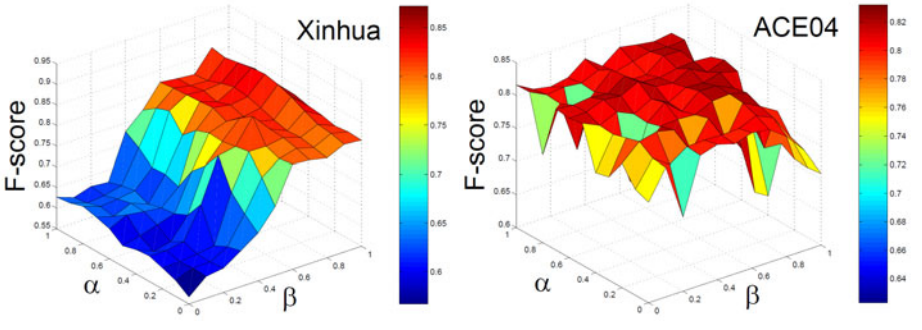

Fig. 4. α, β tuning under a specific γ in *Xinhua* (left) and *ACE*04 (right)

Table 3. Parameter tuning in full LGM

	γ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Xinhua	α_{best}	0.2	0.3	0.3	0.7	0.5	0.3	0.1	**0.3**	0.8	0.3	0.1
	β_{best}	0.6	0.6	0.7	0.6	0.5	0.6	0.4	**0.6**	0.5	0.5	0.4
	F-score	0.771	0.746	0.681	0.837	0.802	0.794	0.651	**0.879**	0.613	0.765	0.719
ACE04	α_{best}	0.6	0.1	0.4	0.5	0.5	**0.6**	0.1	0.2	0.4	0.7	0.8
	β_{best}	0.5	0.6	0.5	0.4	0.4	**0.5**	0.7	0.4	0.7	0.6	0.6
	F-score	0.814	0.767	0.685	0.600	0.784	**0.823**	0.752	0.571	0.629	0.748	0.791

features are used in LGM extraction: (1) context similarity; (2) distance restriction; (3) named entity influence; (4) temporal proximity. Among them, features (2)(3)(4) influence the significance of a sentence. We tried different combinations of these weights with similarity. *LGM-D* means LIs growth model with only distance restriction, *LGM-DN* denotes LIs growth model with distance restriction and named entity influence, *LGM-DT* includes both distance restriction and temporal proximity and finally the full LGM, *LGM-F*, takes all three features into consideration. Within all these measures, the four term weighting strategies, $tf, tf.idf, tf.isf$ and $dtf.isf$ are utilized to see if they bring any benefits. Due to restricted page limits, we present detail results from *Xinhua* in Table 4 and overall performance for both datasets in Fig.5.

Table 4. Performance Evaluation for different models and term weightings in *Xinhua*

	$\overline{p_{sent}}$				$\overline{r_{sent}}$				p_{event}				r_{event}			
	tf	tf.idf	tf.isf	dtf.isf	tf	tf.idf	tf.isf	dtf.isf	tf	tf.idf	tf.isf	dtf.isf	tf	tf.idf	tf.isf	dtf.isf
Baseline-1	0.46	0.39	0.48	0.48	0.31	0.27	0.30	0.31	0.20	0.22	0.28	0.29	1.0	1.0	1.0	1.0
Baseline-2	0.53	0.51	0.56	0.54	0.56	0.50	0.52	0.59	0.23	0.22	0.27	0.31	0.89	0.91	0.90	0.93
LGM-D	0.77	0.69	0.73	0.80	0.62	0.61	0.63	0.68	0.68	0.60	0.71	0.70	0.77	0.71	0.74	0.76
LGM-DN	0.85	0.82	0.87	0.87	0.69	0.63	0.65	0.71	0.67	0.64	0.67	0.68	0.69	0.65	0.69	0.70
LGM-DT	0.79	0.77	0.80	0.84	0.84	0.79	0.83	0.83	0.80	0.78	0.81	0.84	0.79	0.78	0.79	0.80
LGM-F	0.85	0.84	0.87	0.89	0.86	0.81	0.87	0.89	0.84	0.82	0.87	0.87	0.81	0.82	0.84	0.87

From Table 4 we can see different behaviors of the two LIs extraction frameworks. *Baseline*-1 and *Baseline*-2 show obvious weakness in LIs extraction. The reason for this phenomenon is most likely that TextTiling is not designed for event-oriented purpose but for expository texts, so it performs especially dreadful in event level precision because most LIs detected by it are not events. However, it discards no sentences and so recall is extremely high. TTM performs better in that it takes temporal expressions into account which are proved to be valuable. Event-level recall for TTM drops slightly. By comparing four varieties of LGM, we find generally considering all features brings the maximum benefits. The distance restriction is reasonable and brings better results compared to two baselines. The entity influence is relatively limited and it hardly improves sentence-level recall. We believe it is due to the incompleteness of WordNet which has not been updated for a long time but news happens anytime. The correlated entities cannot find a path to each other in WordNet and this probably leads to

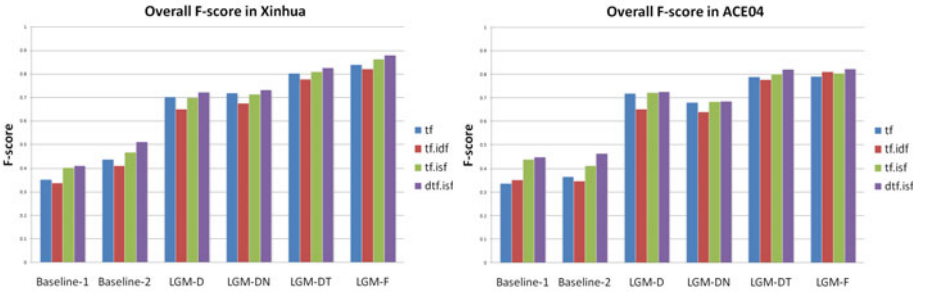

Fig. 5. F-score performance comparison in $Xinhua$ and $ACE04$

the failure of relevant entity influence. Recall is greatly enlarged by the effect of temporal proximity and the importance of temporal information is reconfirmed.

The four term weighting strategies have different results as well. Generally $tf.idf$ performs worst among all these strategies. The reason may be that our evaluation metrics are based on local contexts so that introduction of global information from the whole collection may cause bias. There do not exist the situation that one always prevails over the others for the rest three weighting methods but at most times our innovative $dtf.isf$ beats the other two. We assume this outcome can be ascribed to the local context consideration in document D as well as term distinguishing from sentence level pattern, like $tf.isf$. Therefore it is useful to discover uncommon terms and raise their weights.

6 Conclusion

In this paper, we build a framework to address the novel problem of event recognition from news webpages. We implement two models of LIs extraction for event recognition, utilizing multiple features, either semantical or syntactical. We then provide evaluation metrics to measure the effectiveness of event extraction from news articles and conduct the evaluation methods to two real world datasets. According to the results, among all features we used, distance restriction would be helpful compared with only context similarity utilized in baselines. Unexpectedly, relevant entity influence does not benefit much. Temporal information is proved to be quite essential in LIs extraction task and generally the mixture of all four features brings the best performance in balance.

In the future, our research can be directed to find a substitution of WordNet, such as Wikipedia which is more frequently renewed and thus, up to date. What is more, the hierarchical structure of language thesaurus is not yet used. We treat relationships between adjacent entities equally while in fact they should be distinguished. With structural information and a better content organization of the thesaurus, the effect of relevant entities might be improved.

Acknowledgments. This work is partially supported by NSFC Grant No. 60933004, National Key Technology R&D Pillar Program in the 11th Five-year Plan of China (Research No.: 2009BAH47B00) and the Open Fund of the State Key Laboratory of Software Development Environment Grant No. SKLSDE-2010KF-03, Beihang University. We also thank for the discussions with Pan Gu.

References

1. Allan, J., Papka, R., Lavrenko, V.: On-line new event detection and tracking. In: SIGIR 1998, pp. 37–45 (1998)
2. Banerjee, S., Rudnicky, I.A.: A TextTiling based approach to topic boundary detection in meetings. In: Ninth International Conference on Spoken Language Processing, pp. 57–60 (2006)
3. Bestgen, Y.: Improving text segmentation using latent semantic analysis: A reanalysis of choi, wiemer-hastings, and moore (2001). Comput. Linguist. 32(1), 5–12 (2006)
4. Bestgen, Y., Vonk, W.: The role of temporal segmentation markers in discourse processing. Discourse Processes 19(3), 385–406 (1995)
5. Bestgen, Y., Vonk, W.: Temporal adverbials as segmentation markers in discourse comprehension. Journal of Memory and Language 42(1), 74–87 (1999)
6. Fukumoto, F., Suzuki, Y.: Detecting shifts in news stories for paragraph extraction. In: Proceedings of the 19th International Conference on Computational Linguistics, pp. 1–7 (2002)
7. Fukumoto, F., Suzukit, Y., Fukumoto, J.: An automatic extraction of key paragraphs based on context dependency. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, pp. 291–298 (1997)
8. Grimes, J.: The thread of discourse. Mouton De Gruyter (1975)
9. Hearst, M.: A quantitative approach to discourse segmentation. Computational Linguistics 23(1), 33–64 (1997)
10. Hearst, M.A.: Multi-paragraph segmentation of expository text. In: Proceedings of the 32nd Meeting on Association for Computational Linguistics, pp. 9–16 (1994)
11. Hearst, M.A.: Texttiling: segmenting text into multi-paragraph subtopic passages. Comput. Linguist 23(1), 33–64 (1997)
12. Hearst, M.A., Plaunt, C.: Subtopic structuring for full-length document access. In: SIGIR 1993, pp. 59–68 (1993)
13. Jiang, J., Zhai, C.: Extraction of coherent relevant passages using hidden markov models. ACM Trans. Inf. Syst. 24(3), 295–319 (2006)
14. Misra, H., Yvon, F., Jose, J.M., Cappe, O.: Text segmentation via topic modeling: an analytical study. In: CIKM 2009, pp. 1553–1556 (2009)
15. Ponte, J., Croft, W.: Text segmentation by topic. In: Research and Advanced Technology for Digital Libraries, pp. 113–125 (1997)
16. Salton, G., Singhal, A., Buckley, C., Mitra, M.: Automatic text decomposition using text segments and text themes. In: HYPERTEXT 1996, pp. 53–65 (1996)
17. Van Mulbregt, P., Carp, I., Gillick, L., Lowe, S., Yamron, J.: Text segmentation and topic tracking on broadcast news via a hidden Markov model approach. In: Fifth International Conference on Spoken Language Processing, pp. 2519–2522 (1998)
18. Xie, L., Zeng, J., Feng, W.: Multi-scale texttiling for automatic story segmentation in Chinese broadcast news. In: Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., Zhou, G. (eds.) AIRS 2008. LNCS, vol. 4993, pp. 345–355. Springer, Heidelberg (2008)