Local Graph Matching with Active Learning for Recognizing Inference in Text at NTCIR-10

Tsuyoshi Okita
Dublin City University
tokita@computing.dcu.ie

ABSTRACT

This paper describes the textual entailment system developped at Dublin City University for participation in the textual entailment task in NTCIR-10. Our system is a local graph matching-based system with active learning: we explore reducing the unknown words and unknown named-entities, incorporating meaning in parentheses / rhetorical expressions / semantic roles, and employing text understanding technique using simple logic. We deploy an additional feature of language model from deep learning. Our result was 80.49 for macro F1 score, 84.95 for precision for the positive entailment, and 79.95 for recall for negative entailment.

Categories and Subject Descriptors

I.2.7 [**Computation and Language**]: natural language processing—*textual entailment*

General Terms

Languages

Keywords

Textual Entailment, Text Understanding, Relation Extraction, Paraphrase

Team Name: DCUMT
Task: BC subtask

1. INTRODUCTION

This paper describes the textual entailment system developed at Dublin City University for participation in the textual entailment task in NTCIR-10 [40]. A textual entailment task addresses the variability of semantic expression whether the same meaning can be expressed by or inferred from different texts [13]. More formally, let us call a pair of text expressions T ("Text") and H ("Hypothesis") where the entailing side is T and the entailed side is H. A textual entailment task is to judge for a given pair (T,H) whether T entails H or not: T entails H if a human reading T would infer that H is most likely true. The variability of semantic expressions is a common theme in various applications in NLP including Question Answering (QA), Information Extraction (IE), summarization, and machine translation (MT). A textual entailment task typically involves a wide range of NLP tools in the domain of semantics and various approaches have appeared.

A graph matching approach [19] is a classical approach. Sentences are represented as normalized syntactic dependency graphs and entailment is approximated with an alignment between the graph representing the hypothesis and a portion of the corresponding graph(s) representing the text. Unfortunately, this approach is known to have three problems: (1) other material in the text will not affect the validity of the match since it assumes the upward monotonicity, (2) dropping a restrictive modifier does not preserve entailment in a negative context since the search is based on global features, and (3) it has the inherent problem that alignment and entailment determination is confounding. MacCartney et al. [23] instead employ typed dependency graphs, partial alignment between the typed dependency graphs representing the hypothesis and the text, and a decision of entailment. Mirkin et al. [26] deploy the partial graph which they call a subsentential textual entailment. Bentivogli et al. [4] focus on coreference in the context of discourse. First, the topic is often among coreferent (In event coreference resolution, event is often among the topic) and, hence, anaphoric NPs seem to be important which are related to the second subproblem in a coreference resolution task consisting of three subproblems: named entity recognition, anaphoricity determination, and coreference element detection [38]. Anaphoricity determination is to extract deterministic anaphoric NPs from the text in the case of English [2]. Second, the resolution of time / space references are in itself necessary to determine whether T entails H.

Our approach is a graph matching-based approach similar with Haghighi et al. [19], with the similar modification by MacCartney et al. [23] or Mirkin et al. [26]. Especially, we explore various preparation methods in its preprocessing in order to augment the indispensable data for graph matching: hence, this is active learning. The characteristics of our system lies in these three. First, our system prepares indispensable additional data with preprocessing which reduces the unknown words and unknown named-entities, incorporates meaning in parenthesis / rhetorical expressions / semantic

roles, and prepares the (simple) text understanding capability in graph matching. Second, in psychology and color comparison [15], the comparison between two with a big difference tends to be avoided. This is called a principle of just noticeable difference. Followed by this, our approach tries not to compare T' and H in big difference. We tried to decide whether this T entails H or not within a single aspect with very small difference. Third, the context-dependent language model is developed in the context of deep learning which became popular recently [3, 37, 25]. This context-dependent language model is reported to be of less perplexity than the state-of-the-art language model. As is shown by Collobert [11], this context-dependent language model facilitated the implementation of Semantic Role Labeller (SRL), named-entity recognizer, and parser. As far as we know, there have been no reports as this being applied to a textual entailment task.

The remainder of this paper is organized as follows. Section 2 describes the overview of our systems. In Section 3, we describe the linguistic and shallow preprocessing step for preparing indispensable data for graph matching, while in Section 4 we describe the determination step of graph matching. Our experimental results are presented in Section5. We conclude in Section 5.

2. OVERVIEWS

Review of Graph Matching Model. As is similar with Haghighi et al. [19], we represent text of T and H as a graph in the following way. First, T and H are represented as a dependency tree using the modified version of Collins' head propagation rules [10], i.e. main verbs are placed at the head of sentences. Second, the dependency nodes such as collocations and named-entities are collapsed. Note that collocations include verbs and their adjacent particles. Third, certain dependencies such as modifying prepositions are folded. Fourth, the graph representation is augmented by Propbank-style semantic roles. Each predicate adds an arc labeled with the appropriate semantic role to the head of the argument phrase. Modifying phrases are labeled with their semantic types.

The summary of the graph matching model introduced by Haghighi et al. [19] is as follows. Let H denote hypothesis graph, T denote a text graph, M denote a mapping from the vertices of H to those of T, $M(v)$ denote the match in T for vertex v in H , and $Cost(M)$ be the cost of matching M. When M is a set of matching, the cost of matching H to T is defined as in (1):

$$\text{MatchCost(H,T)} = \min_{M \in \mathcal{M}} \text{Cost}(M) \quad (1)$$

where $\text{Cost}(M)$ is given by a convex mixture of the node and relational match costs as in (2):

$$\text{Cost(M)} = \alpha \text{NodeCost(M)} + (1-\alpha)\text{RelCost(M)} \quad (2)$$

where NodeCost(M) denotes a node cost, and RelCost(M) denotes relational match cost. Let $NodeSub(v, M(v))$ be a model for substituting node v for $M(v)$. Then, node cost is represented as in (3):

$$\text{NodeCost}(M) = \frac{1}{Z} \sum_{v \in H_v} w(v) NodeSub(v, M(v)) \quad (3)$$

where $w(v)$ denotes the weight for node v, and Z ($=\sum_{v \in H_v} w(v)$) denotes a normalization constant. Similarly, let $PathSub(e, \phi_M(e))$ be a model for assessing the cost of substituting a direct relation e for $\phi_M(e)$ under the matching. Relation cost is represented as in (4):

$$\text{RelCost}(M) = \frac{1}{Z} \sum_{v \in H_v} w(e) PathSub(e, \phi_M(e)) \quad (4)$$

where $w(e)$ denotes a edge cost, and Z ($=\sum_{v \in H_v} w(e)$) denotes a normalization constant. In sum, this model yields T entails H when $\text{MatchCost}(H, T)$ is low, and otherwise T does not entail H.

Our Model. Now in our model, we introduce the locality to the graph matching algorithm. Let T_j be subgraph of T and H_i be subgraph of H. For example, when T consists of multiples of sentences, T_j may be a simple sentence. Let $H_i \approx T_j$ denote that H_i is close enough to T_j. Avoiding to assess the cost globally, we decompose $\text{Cost}(H, T, M_{(i,j)})$ with a set of $\text{LocalCost}(H_i, T_j, M_{(i,j)})$ where each decomposed subgraph which includes a close pair of H_i and T_j which satisfies $H_i \approx T_j$. Hence, the modified version of the MatchCost(H,T) can be written as in (6):

$$\text{MatchCost(H,T)} = \min_{M \in \mathcal{M}} \text{Cost}(H, T, M_{(i,j)}) \quad (5)$$

$$\text{Cost}(H, T, M_{(i,j)}) = \sum_{H_i \approx T_j} \text{LocalCost}(H_i, T_j, M_{(i,j)}) \quad (6)$$

Note that this $H_i \approx T_j$ corresponds to the principle of the just noticeable difference in psychology [15]. Although this indicates that for given T and H it may not be possible to find out such (a set of) H_i and a set of T_j. Other note is that if H_i only refers a subset of T, it may not need to consider other part of T. This means that we may not need to iterate all of subset of T to compare H_i. This also means that we treat this as if a set of H_i and T_i is almost mutual exclusive and only a couple of pairs of H_i and T_i are active in practice.

Deep / Shallow Linguistic Preprocessing Step. As is written in Haghighi et al. [19], the node and edge substitution models uses some linguistic preprocessing such as Part-Of-Speech (POS) tagger, Latent Semantic Analysis (LSA), Wordnet, stemmer, and so forth. In our view, the information obtained by those linguistic preprocessing are fairly limited and basically passive processing. We extend this and actively prepare the presumably indispensable information to calculate a correct MatchCost(H,T). In this reason, we call this active learning. The elements of active learning is other than the conventional linguistic methods: parsing, morphological analysis, noun phrase extraction (and named-entity extraction), and so forth. We do determination of unknown noun phrases and unknown named-entities, rhetoric detection, relation extraction, text understanding, paraphrasing, time / space coreference resolution. Especially, text understanding is to actively examine the elements in text and infer using a simple logic. We also training of context-dependent language models.

Based on these preprocessing steps, T and H will have various semantic annotations with the representation as dependency graphs. One more processing is subject alignment and transformation: first, the subject in H is aligned to the phrase T_s in the partial fragment in T. If T_s is not the

subject in the original T, T is transformed with T_s as the subject. For example, suppose that we are given the following T and H:

- T: 自激漏（じげきろう）は、1434 年に中世李氏朝鮮の科学者、蒋英実が作った水時計である。
- H: 蒋英実は中世李氏朝鮮の科学者である。

In this case, the subject alignment connects '蒋英実' in H with '蒋英実' in T. Then, the transformation yields the graph containing several subtrees. If we extract such subtrees, this becomes the following four subtrees in T.

- T_1: <person>蒋英実</person>は、<job>科学者</job>である。
- T_2: <person>蒋英実</person>は、<country>中世李氏朝鮮</country>の<job>科学者</job>である。
- T_3: <person>蒋英実</person>は、<time>1434 年</time>に<object>水時計</object>を作った。
- T_4: coordination [<phrase>自激漏</phrase>, <phrase>じげきろう</phrase>]
- H: <person topic='Y'>蒋英実</person>は<country>中世李氏朝鮮</country>の<job>科学者</job>である。

Hence, the graph matching algorithm eventually calculates the cost mostly between H and T_2.

3. DEEP / SHALLOW LINGUISTIC PREPROCESSING STEP

The deep / shallow linguistic preprocessing step modifies the original structures of H and T in order to provide H_i and T_j which are subset of H and T with corresponding features in the classification step. Using the equation (5) and (6), our algorithm makes the size of the source and the target sentences shrinked in order that we can compare the H_i and T_j where H_i and T_j are close enough and where other LocalCost(H_i,T_j,$M_{(i,j)}$) can be considered to be zero (although this is not always the case). In this process, the structure of texts are actively investigated in two directions: (1) make T from complex / compound sentences into simple sentences and (2) make the form of T simplified with considering the easier match with H. At the same time, the feature extraction are actively proceeded in order to help the simplification of T. We call this mechanism as *active learning* since the features used in the standard SVM are not modified but are globally evaluated. Note that although it is often the case that active learning let increase the training data, the active learning here let decrease the substructure of training data and let extract the features according to this dynamical substructure. The conversion of subjects relations (or subject alignment) of T_i towards H_i is done in this process, as well as replacement of unknown words / named-entities, text understanding and other linguistic preprocessing which are enumerated in this chapter. In the experiments, we used various deep / shallow linguistic preprocessing tools as well as resources, which are shown below.

- Morphological analyzer: JUMAN [22].
- Dependency parser: KNP [21].
- Named-entity recognizer, NLTK [5], MALLET [24].
- Paraphrase generator [9]; ngram-Hidden Markov Model (HMM) language model [6]; bootstrap method ("X deploy Y").
- Wordnet [7], Wiki, monolingual corpora, and parallel corpora.
- Internet search engine: Google; Yahoo.
- Deep learning component: context-dependent recurrent neural network language model [25], ngram-HMM language model [6];

Unknown Words (OOV Words). In textual entailment task, it is expected that unknown words or out-of-vocabulary words (OOV words) have negative effects whatever the steps are. When the system encounters unexpected POS sequences or doubtful sequences, based on the heuristic to derive candidate phrases based on the knowledge of case particles, our system searches the Internet resources and obtain the (rank 1 to rank 100) results. We use the heuristic that it is often possible to segment phrases by the knowledge of Japanese case particles and that of other segments which are more certain. If the results include titles, it is often likely that the candidate phrase is identified. We make a correction of the phrase boundaries. (Upon recognizing a mistake, we run again the morphological analyzer and parser with replacing some easier candidates and later replace back.)

Unknown Named Entities (Multi-Word Expressions). The unknown named-entities of the proper nouns, such as person name, company names, and titles, may also considerably decrease the overall performance of the system. We use the similar heuristic as unknown words to find a possibly correct named entities (Multi-Word Expressions). Note that name can be written in various ways. For example, Leonald Da Vinci is equivalent with "Da Vinci", "Mr. Leonald Da Vince", "Leonald", and so forth.

Parenthesis and Quotation. There are several different meanings of parenthesis and quotation. It is expected that the text within parenthesis are equivalent expression of the items before the parenthesis. Such symbols include "『』", "（）", "「」", "“”", "〝〟", and so forth. Sometimes the (syntactic) adjuncts are also better to be considered as one named-entity. For example, "第 8 回『このミステリーがすごい!』大賞" can be considered as one entity rather than only considering "『このミステリーがすごい!』".

- Equivalent expression:
  - 1993 年 (平成 5 年), 世界保健機関 (WHO)
- Explanation:
  - ラフテー（東坡肉が元祖）
- More specific / generic explanation:
  - 携帯の一般的な入力方法（トグル打ち・マルチタップ）
- Explanation of attribute (birthday, organization, and so forth):

– ロバート・エドワード・ターナー三世（1938 年 11 月 19 日 -）

- Specification of segmentation:
  – 後に、『新世紀エヴァンゲリオン』における「Project EVA」

Noisy Characters. The text between parentheses and quotes may include noisy characters. For example, “第 8 回『このミステリーがすごい!』大賞” includes “!”. “略称”, “例：”, “「」”, and so forth.

- “『』”, “!”, “-”.
  – 第 8 回『このミステリーがすごい!』大賞
- “略称”
  – 独立行政法人情報処理推進機構（じょうほうしょりすいしんきこう、Information-technology Promotion Agency Japan、略称「IPA」）

Hypergraph Representation of T. We identify the dependency structure of T and H by dependency parser. We formulate a hypergraph. For example, T can be decomposed into several sentences $\{T_1, \ldots, T_n\}$, which is shown in the following example.

- T:せたまるは、Suica と同じく非接触式 IC カード通信技術 FeliCa を採用しているが、カード端の切欠きはなく、IC チップ内のファイル構造も Suica とは一部異なる。
- T_1: せたまるは、Suica と同じく非接触式 IC カード通信技術 FeliCa を採用している。
  – T_{11}: (せたまるは、非接触式 IC カード通信技術 FeliCa を採用している。)
  – T_{12}: (Suica は、非接触式 IC カード通信技術 FeliCa を採用している。)
- T_2: せたまるは、カード端の切欠きはない。
- T_3: せたまるは、IC チップ内のファイル構造も Suica とは一部異なる。

Note that T_1 is further decomposed into T_{11} and T_{12}.

Subject Alignment. The subject can be freely modified according with H. In the following examples, the focus of H is on ’ナチス・ドイツ’ and its counter part ’ポーランド’. T is decomposed according to this information: that is, (1) ’ナチス・ドイツ’は、’ポーランド’に侵略した, (2) ’ナチス・ドイツ’は、’ポーランド’をホロコーストの舞台とした, and (3) ナチス・ドイツとポーランドとの間では、共同研究が進んでいなかった.

- T:ナチス・ドイツが侵略し、ホロコーストの主な舞台になったポーランドとの間では、東西冷戦の影響で共同研究が進んでいなかった。
- H:ナチス・ドイツはポーランドに侵略した。

Coordination. At the same time when we derive a hypergraph, we obtained the relation of coordination if there is.

1. T': 天才といえば古くはレオナルド・ダ・ヴィンチ、近代に入ってはエジソンとアインシュタインと決まっている。
2. coordination: レオナルド・ダ・ヴィンチ, エジソン, アインシュタイン

Coreference Resolution / Identification of Nonanaphoric NPs. In coreference resolution, the non-anaphoric definite NPs [2] are often given, but in our context they should be identified in its preparation. This should be also true for relative pronouns, reflexive pronouns, personal pronouns as well. Note that since there is no article in Japanese we have no distinction between whether definite NPs and nondefinite NPs. We identify the nonanaphoric NPs.

Coreference Resolution of Space / Time References. We employ the space and time coreference resolution to identify the fluctuation of space and time expressions.

Text Understanding. A graph-matching-based textual entailment [19, 23] has limitation in that they will not detect whether T requires to understand the content of H. Suppose that T=’Bob bought a red car from Mary and Tom.’ and H=’Three persons are related to the conversation’. In T, there is no number appeared, but human beings can read this sentence and understand that there are three persons in T. Slightly more difficult example, such as follows, is appeared in development set.

- T:世界の地震の約１割が日本周辺で起きていて、マグニチュード８を超える巨大地震も２割近くが日本周辺で起きているという。
- H:マグニチュード８を超える巨大地震の約１割が日本周辺で起きている。

In this example, RTE system needs a simple calculation. ’日本周辺の地震’は約１割, ’マグニチュード８を超える地震’は２割. Hence, the superposition of ’マグニチュード８を超える巨大地震’ and ’日本周辺の地震’ is 約 0.2 割 by subtracting ２割-約１割. The system should have the capability of understand the sentence and calculate in this way.

Note that our system only supports very simple operations by first-order logic since it is very difficult to simulate whole the capability of human beings. We categorize (1) time description (“when”), (2) location description (“where”), (3) who description (“who”), (4) what description (“what” or “which”), (5) size description (“how” followed by an adjective), (6) number description (“how many”), and (7) why description (“why”). Note that although there is no question sentence in corpus of textual entailment it would be possible that we assume that a hypothesis sentences can be categorize one of these which answers to the question which is hidden behind the scene.

Text Understanding Time Inference. It is often that the time description may need to deal with the time information given in T. The following example needs to understand that a week after 31/May/2012 means 6/June/2012 and hence 2/June/2012 is before this date.

- T: The explosion was happened on 31/May/2012 and it made the airplane delayed for a week.
- H: The airplane on 2/June/2012 was not delayed.

Text Understanding Location Inference. Similarly, location description may need to deal with the location information given in T. In the following example, in order to judge the entailment of this example, it is required to understand simple location knowledge: 'Japan' ⊆ 'World'.

- T: There are thousands of earthquake in the world.
- H: The number of earthquake in Japan is less than thousands.

Slightly different setting is in scene understanding. In this example, in order to judge the entailment of this example, it is required to understand the location of each object in space: hence, a dog is behind a big building, implicating that a dog is not visible from Tom.

- T: There is a big building in front of Tom and there is a dog behind the building.
- H: Tom cannot see a dog.

Text Understanding Number Inference. In the following, it is required to count there are 3 persons appeared in a sentence. In order to do this, it is required a function which counts how many person exist in a sentence.

- T: 天才といえば古くはレオナルド・ダ・ヴィンチ、近代に入ってはエジソンとアインシュタインと決まっている。
- coordination: レオナルド・ダ・ヴィンチ, エジソン, アインシュタイン

Hypernym and Antonym. This example is an usual situation for many literature. In a word level, if there is some difference in terms of the level of abstraction in two words, i.e. 中高年 and 中高生, it is required to judge whether 中高年 is a hypernym of 中高生 or these two does not have such relationships. Such relationships in word can be judge using lexical resources such as (Japanese) WordNet [7].

Quantifier Detection. In English sentence, a quantifier, such as 'all' and 'every', needs to be examined in order to grasp the correct meaning. These are detected by the predefined vocabularies.

- 刑事訴訟では疑わしきは罰せずという推定無罪が原則であるため、保険金殺人の事件が殺人罪として有罪にならないこともある。

Rhetoric Detection. If the sentence includes rhetoric, such as metaphor, prosopopoeia, and the idiomatic expression such as the four-word Kanji (i.e. "温故知新" and "南船北馬"), this may prevent the similarity-based matching approach. It is often that the title becomes rhetoric, such as in the case of "『Hey Hey おおきに毎度あり』", in the sense that even if the meaning in the title matches with the surrounding meaning, it does not mean that T entails H. The text within "『』" should be considered to be a different layer of meaning.

- 『Hey Hey おおきに毎度あり』は，タイトルが表すように、歌詞は全編関西弁で描かれ、メロディ部分はほぼ台詞であったりという変わった作風である。ト・ト」（全軍突撃せよ）及び「トラトラトラ」（奇襲ニ成功セリ）が淵田中佐機から打電されたことで知られる。

4. DETERMINATION STEP

The determination step judges the similarity of the T_i and H_i by the SVM classification [8, 12] where T_i and H_i are the possible correspondent fragments. As is mentioned above, the feature extraction for the SVM classification algorithm are applied for the selected H_i and T_j. We used L1-loss function with Radial Basis Function (RBF) kernel where C and γ were determined by cross-validation. Major features which we used in our system are described below.

Lexical Entailment / Hyponymy Relations / Antonymy Relations / Location Relations / Adjective Gradation Features. These features are the same as [14] and [23]. Note that depending on the deep / shallow linguistic preprocessing, hyponymy relations and antonymy relations are exchanged.

Modality / Polarity / Factivity Features. These features capture the contexts which reverse or block monotonicity [23] where these are often marked by the presence or absence of linguistic markers. Modality feature capture modal reasoning where possibility will not entail actuality. Factivity feature

Adjunct Feature. This feature suggests the dropping or adding of syntactic adjuncts moving from T to H [23].

Quantifier Features. These features captures entailment relations among sentences involving quantification [23].

Semantic Role Matching Feature. This feature indicates whether the corresponding semantic role relations are equivalent or not. As with this feature, some pair of features are preprocessed to give true or false beforehand.

Parenthesis and Quotation Features. These features indicate the presence or absence of possible equivalent expressions. This enables the similarity matching with the expression among parenthesis and quotation.

Noisy Character / Rhetoric Feature. These features suggest to drop the corresponding fragments from the similarity matching.

Time / Date / Number Features. The presence of these features can be preprocessed by coreference resolution of space / time references (or some localization software) which will detect different form of equivalent expressions. These features are often preprocessed beforehand whether they are true of false.

Text Understanding Features. Classification can only capture the similar expressions between T_i and H_j. As is mentioned in Chapter 3, when H_j requests some capability of text understanding of T_i, this feature would suggest some basic inference results in the deep / shallow linguistic preprocessing. This enables a judge whether T_i can be entailed H_j. Note that the capability of these features are limited in time, location, and number and in very basic case.

Content Length Feature. If H contains more information than T, this can be immediately decided that T does not entail to H.

Deep Learning Language Model Feature. Context- dependent language model feature is derived by context-dependent recurrent neural network language model [25] and ngram-HMM language model [6].

Genre ID Feature. Genre ID feature is derived by Latent Dirichlet Allocation (LDA) [24].

5. EXPERIMENTAL SETTINGS AND RESULTS

The statistics of development and test set for textual entail BC task is shown in 5. The result by our approach is shown in Table 5. The Macro F1 score was 80.49. The precision for yes entailment was high, while the recall for no entail was high. As is indicated by the row of our submission, our textual entailment system gave output of 'yes' with much smaller number than the correct answer, while it gave output of 'no' in larger number.

	Yes	No	Total
JA devset	240	371	611
JA testset	256	354	610
Our submission	206	404	610

	DCUMT		
MacroF1	80.49		
Accuracy	81.64		
Y-F1	75.76	N-F1	85.22
Y-Prec	84.95	N-Prec	79.95
Y-Rec	68.36	N-Rec	91.24

6. CONCLUSION

We participated in textual entailment task for NTCIR-10. Our system is a variant of graph matching-based system with additional capability of reducing the unknown words and unknown named-entities, incorporating meaning in parentheses / rhetorical expressions / semantic roles, understanding of simple logic. We used an additional feature of language model from deep learning. Our result was 80.49 for macro F1 score, 84.95 for precision for the positive entailment, and 79.95 for recall for negative entailment. The reason for high precision for the positive entailment may be due to the fact that we tried to determine the entailment only when the distance between T' and H becomes small depending on deep / shallow linguistic preprocessing and determination.

There are several avenues for further work. First, we applied Japanese textual entailment. We would like to extend this to English textual entailment as well as cross-lingual textual entailment. Although the direct application would be the evaluation task in SMT, it is possible to apply this technique to SMT components such as word alignment task [27, 31, 30] and decoding task [34, 35]. This would lead to the (deep) semantically informed SMT. Second, we would like to use the testset which has not much unknown words in order to specify the effect of deep learning architecture although our usage is limited in paraphrasing which is similar to [28, 29] which applied to system combination task [35]. This way of building SMT components is related to unsupervised learning of sentence or language model [37]. This line of research would lead to the (shallow) semantically informed SMT [28, 32, 33].

7. ACKNOWLEDGMENTS

We thank John Judge for proof-reading of Introduction. This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (`http://www.cngl.ie`) at Dublin City University. We would also like to thank the Irish Centre for High-End Computing.

8. REFERENCES

[1] M. J. Beal. Variational algorithms for approximate bayesian inference. *PhD Thesis at Gatsby Computational Neuroscience Unit, University College London*, 2003.

[2] D. Bean and E. Riloff. Unsupervised learning of contextual role knowledge for coreference resolution. *In Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL 2004)*, 2004.

[3] Y. Bengio, R. Ducharme, and P. Vincent. A neural probabilistic language model. *In Proceeding of the Advances in Neural Information Processing Systems 13 (NIPS00) –MIT Press*, 2001.

[4] L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, M. L. Leggio, and B. Magnini. Considering discourse references in textual entailment annotation. *5th International Conference on Generative Approaches to the Lexicon (GL 2009)*, 2009.

[5] S. Bird, E. Klein, and E. Loper. Natural language processing with python— analyzing text with the natural language toolkit. *O'Reilly Media*, 2009.

[6] P. Blunsom and T. Cohn. A hierarchical pitman-yor process hmm for unsupervised part of speech induction. *In Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-11)*, 2011.

[7] F. Bond, H. Isahara, S. Fujita, K. Uchimoto, T. Kuribayashi, and K. Kanzaki. Enhancing the japanese wordnet. *In Proceedings of the 7th Workshop on Asian Language Resources (in conjunction with ACL-IJCNLP 2009)*, 2009.

[8] B. E. Boser, I. M. Isabelle M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. *In Proceedings of the 5th Annual ACM Workshop on COLT*, pages 144–152, 1992.

[9] C. Callison-Burch, D. Talbot, and M. Osborne. Statistical machine translation with word- and sentence-aligned parallel corpora. *In Proceedings of the*

42th Annual Meeting of the Association for Computational Linguistics (ACL-04), pages 175–182, 2004.

[10] M. Collins. Head-driven statistical models for natural language parsing. *PhD Dissertation (University of Pennsylvania)*, 1999.

[11] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. *International Conference on Machine Learning, ICML*, 2008.

[12] N. Cristianini and J. Showe-Taylor. Introduction to support vector machines. *Kyoritsu Publisher*, 2005. Translated by Tsuyoshi Okita.

[13] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. *In Proceedings of the First PASCAL Recognizing Textual Entailment Workshop*, 2006.

[14] M.-C. de Marneffe, T. Grenager, B. MacCartney, D. Cer, D. Ramage, C. Kiddon, and C. D. Manning. Aligning semantic graphs for textual inference and machine reading. *AAAI Spring Symposium at Stanford*, 2006.

[15] D. A. Forsyth and J. Ponce. Computer vision. *Kyoritsu Publisher*, 2007. Translated by Tsuyoshi Okita.

[16] J. V. Gael, A. Vlachos, and Z. Ghahramani. The infinite hmm for unsupervised pos tagging. *The 2009 Conference on Empirical Methods on Natural Language Processing (EMNLP 2009)*, 2009.

[17] Z. Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):9–42, 2001.

[18] S. Goldwater, T. L. Griffiths, and M. Johnson. Contextual dependencies in unsupervised word segmentation. *In Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics (COLING-ACL06)*, pages 673–680, 2006.

[19] A. Haghighi, A. Ng, and C. D. Manning. Robust textual inference via graph matching. *In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2005)*, 2005.

[20] R. Kneser and H. Ney. Improved backing-off for n-gram language modeling. *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 181–184, 1995.

[21] S. Kurohashi and M. Nagao. KN Parser : Japanese Dependency/Case Structure Analyzer. *In Proceedings of The International Workshop on Sharable Natural Language Resources*, pages 48–55, 1994.

[22] S. Kurohashi, T. Nakamura, Y. Matsumoto, and M. Nagao. Improvements of Japanese Morphological Analyzer JUMAN. *In Proceedings of The International Workshop on Sharable Natural Language Resources*, pages 22–28, 1994.

[23] B. MacCartney, T. Grenager, M.-C. de Marneffe, D. Cer, and C. D. Manning. Learning to recognize features of valid textual entailments. *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2006)*, 2008.

[24] A. K. McCallum. Mallet: A machine learning for language toolkit. *http://mallet.cs.umass.edu*, 2002.

[25] T. Mikolov and G. Zweig. Context dependent recurrent neural network language model. *MSR-TR-2012-92*, 2012.

[26] S. Mirkin, I. Dagan, and E. Shnarch. Evaluating the inferential utility of lexical-semantic resources. *In Proceedings of Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009) Student Research Workshop*, 2009.

[27] T. Okita. Data cleaning for word alignment. *In Proceedings of Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009) Student Research Workshop*, pages 72–80, 2009.

[28] T. Okita. Neural Probabilistic Language Model for System Combination. *In Proceedings of ML4HMT Workshop, collocated with COLING 2012*, 2012.

[29] T. Okita. Joint Space Neural Probabilistic Language Model for Statistical Machine Translation. *arXiv: 1301.3614*, 2013.

[30] T. Okita, Y. Graham, and A. Way. Gap between theory and practice: Noise sensitive word alignment in machine translation. *In Proceedings of the Workshop on Applications of Pattern Analysis (WAPA2010). Cumberland Lodge, Engl and.*, 2010.

[31] T. Okita, A. M. Guerra, Y. Graham, and A. Way. Multi-Word Expression sensitive word alignment. *In Proceedings of the Fourth International Workshop On Cross Lingual Information Access (CLIA2010, co llocated with COLING2010), Beijing, China.*, pages 1–8, 2010.

[32] T. Okita, R. Rubino, and J. van Genabith. Sentence-level quality estimation for MT system combination. *In Proceedings of ML4HMT Workshop, collocated with COLING 2012*, 2012.

[33] T. Okita, A. Toral, and J. van Genabith. Topic modeling-based domain adaptation for system combination. *In Proceedings of ML4HMT Workshop, collocated with COLING 2012*, 2012.

[34] T. Okita and J. van Genabith. DCU Confusion Network-based System Combination for ML4HMT. *Shared Task on Applying Machine Learning techniques to optimising the division of labour in Hybrid MT (ML4HMT-2011, collocated with LIHMT-2011)*, 2011.

[35] T. Okita and J. van Genabith. Minimum Bayes Risk Decoding with Enlarged Hypothesis Space in System Combination. *In Proceedings of 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2012)*, pages 40–51, 2012.

[36] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[37] H. Schwenk and J.-L. Gauvain. Connectionist language

modeling for large vocabulary continuous speech recognition. *In International Conference on Acoustics Speech and Signal Processing*, pages 765–768, 2002.

[38] V. Stoyanov, N. Gilbert, C. Cardie, and E. Riloff. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. *In Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009)*, 2011.

[39] Y. W. Teh. A hierarchical bayesian language model based on pitman-yor processes. *In Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL-06), Prague, Czech Republic*, pages 985–992, 2006.

[40] Y. Watanabe, Y. Miyao, J. Mizuno, T. Shibata, H. Kanayama, C.-W. Lee, C.-J. Lin, S. Shi, T. Mitamura, N. Kando, H. Shima, and K. Takeda. Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10. *Proceedings of the 10th NTCIR Conference*, 2013.

Appendix: Ngram-HMM Language Model

Generative model. Figure 1 depicted an example of ngram-HMM language model [6] in blue (in the center): Hidden Markov Model (HMM) [36, 17, 1] of size K emits n-gram word sequence $w_i, \ldots, w_{i-K+1}$ where $h_i, \ldots, h_{i-K+1}$ denote corresponding hidden states, while the arcs from w_{i-3} to w_i, $\cdots$, w_{i-1} to w_i show the backoff relations appeared in language model smoothing, such as Kneser-Ney smoothing [20] and hierarchical Pitman-Yor LM smoothing [39].

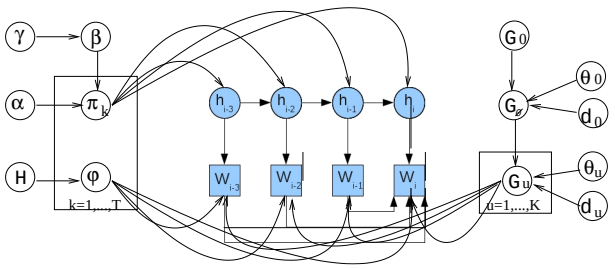

Figure 1: Figure shows the 4-gram HMM language model and generative model.

In the left side in Figure 1, we place one Dirichlet Process prior $\mathrm{DP}(\alpha, \mathrm{H})$, with concentration parameter α and base measure H, for the transition probabilities going out from each hidden state. This construction is borrowed from the infinite HMM [1, 16]. The observation likelihood for the hidden word h_t are parameterized as in $w_t|h_t \sim F(\phi_{s_t})$ since the hidden variables of HMM is limited in its representation power. This is since the observations can be regarded as being generated from a dynamic mixture model [16] as in (7), the Dirichlet priors on the rows have a shared parameter.

$$\begin{aligned} p(w_i|h_{i-1}=k) &= \sum_{h_i=1}^{K} p(h_i|h_{i-1}=k)p(w_i|h_i) \\ &= \sum_{h_i=1}^{K} \pi_{k,h_i} p(w_i|\phi_{h_i}) \end{aligned} \tag{7}$$

In the right side in Figure 1, we place Pitman-Yor prior PY:

$$w_i|w_{1:i-1} \sim \mathrm{PY}(d_i, \theta_i, G_i) \tag{8}$$

where α is a concentration parameter, θ is a strength parameter, and G_i is a base measure. This construction is borrowed from hierarchical Pitman-Yor language model [39].

Inference. We compute the expected value of the posterior distribution of the hidden variables with a beam search [16]. This blocked Gibbs sampler samples the parameters (transition matrix, output parameters), the state sequence, hyper-parameters, and the parameters related to language model smoothing, turn in turn. As is mentioned in [16], this sampler has characteristic in that it adaptively truncates the state space and run dynamic programming as in (9):

$$\begin{aligned} p(h_t|w_{1:t}, u_{1:t}) &= p(w_t|h_t) \\ &\sum_{h_{t-1}:u_t<\pi^{(h_{t-1},h_t)}} p(h_{t-1}|w_{1:t-1}, u_{1:t-1}) \end{aligned} \tag{9}$$

where u_t is only valid if this is smaller than the transition probabilities of the hidden word sequence $h_1, \ldots, h_K$. Note that we use an auxiliary variable u_i which samples for each word in the sequence from the distribution $u_i \sim \mathrm{Uniform}(0, \pi^{(h_{i-1},h_i)})$. The implementation of the beam sampler consists of preprocessing the transition matrix π and sorting its elements in descending order.

Initialization. First, we obtain the parameters for hierarchical Pitman-Yor process-based language model [39, 18].

Second, in order to obtain a better initialization value h for the above inference, we perform the following EM algorithm instead of giving the distribution of h randomly. This EM algorithm incorporates the above mentioned truncation [16]. expected value of the posterior distribution of the hidden variables. For every position h_i, we send a forward message $\alpha(h_{i-n+1:i-1})$ in a single path from the start to the end of the chain (which is the standard forward recursion in HMM). Here we normalize the sum of α considering the truncated variables $u_{i-n+1:i-1}$.

$$\begin{aligned} \alpha(h_{i-n+2:i}) &= \frac{\sum \alpha(h_{i-n+1:i-1})}{\sum \alpha(u_{i-n+1:i-1})} P(w_i|h_i) \sum \\ &\alpha(u_{i-n+1:i-1}) P(h_i|h_{i-n+1:i-1}) \end{aligned}$$

Then, for every position h_j, we send a message $\beta(h_{i-n+2:i}, h_j)$ in multiple paths from the start to the end of the chain as in (10),

$$\begin{aligned} \beta(h_{i-n+2:i}, h_j) &= \frac{\sum \alpha(h_{i-n+1:i-1})}{\sum \alpha(u_{i-n+1:i-1})} P(w_i|h_i) \\ &\sum \beta(h_{i-n+1:i-1}, h_j) P(h_i|h_{i-n+1:i-1}) \end{aligned}$$

This step aims at obtaining the expected value of the posterior distribution. In the M-step, using this expected value of the posterior distribution obtained in the E-step to evaluate the expectation of the logarithm of the complete-data likelihood.