3M2-IOS-3b-2

Utilising Bilingual Lexical Resources for Technical Term Extraction

Panot Chaimongkol[*1] Pontus Stenetorp[*1] Akiko Aizawa[*1*2]

[*1] The University of Tokyo [*2] National Institute of Informatics

Technical Term Extraction (TTE) is the task of detecting mentions of technical terms in scientific texts; it is closely related to Named Entity Recognition (NER). TTE is a stepping-stone to perform automated analysis of scientific texts and is essential for well-established tasks such as information extraction and knowledge retrieval. For NER, annotated resources are commonly coupled with supervised learning methods to produce and evaluate state-of-the-art systems. However, the current lack of annotated resources for TTE hampers further research efforts. To perform a preliminary study we induce annotations by exploiting author keywords assigned to scientific texts. We construct a baseline system by training a Conditional Random Field model and a set of well-established NER features. Furthermore we examine potential benefits of incorporating extra linguistic resources for TTE utilising bilingual dictionary resources. Mere dictionaries, however, are not enough to identify technical terms; various ambiguities must be clarified using information from co-occurrence of words. It is our hypothesis that bilingual dictionaries are promising for disambiguation of meanings by looking at cross-language information. The experiments show that our proposed models with bilingual dictionary features perform slightly better than baseline model.

1. Introduction

1.1 Natural language processing

The amount of data in digital form had reportedly exceeded 1 zettabyte (10^{21} bytes or a billion terabyte) in 2010 and was expected to surpass 1.8 zettabytes in 2011 [GR11]. The number is the result of the 9-fold growth in 5 years, and the growth is expected to continue as more people gain access to digital equipments. A significant portion of this data is conveyed in plain text in natural language which content a computer cannot easily process due to its unstructured and ambiguous nature; a word or sentence can have multiple interpretation which depends heavily on its context and extensive knowledge of the world.

However, the explosion of the size of digital data makes clear that it is beneficial and also necessary to process the data in an automated manner since the amount of data has grown to a level well beyond what humans can reasonable handle using manual means. Natural Language Processing (NLP) is a subfield of Artificial Intelligence (AI) concerning the automated processing of natural languages and thus addresses the need for automated processing of natural language texts. NLP includes tasks such as information extraction, machine translation and question answering.

1.2 Technical term extraction

Information Extraction (IE) is a task in NLP that involves transforming natural language texts into a form that enables a computer to process the information embedded in the natural language text. In short, IE is the task that makes a computer able to interpret the information contained in natural language texts.

IE is a complex task that is divisible into many sub-tasks. Named Entity Recognition (NER) is a sub-task which serves as a stepping-stone for deeper analysis in other sub-tasks. NER is the task of identifying named entities, anything that can be referred with proper names, in a given text. It is a critical sub-task in IE since named entities are very likely the main objects involved in statements contained in text.

Among various types of digital texts, the number of scientific writings also increases significantly. In this paper, we focus on Technical Term Extraction, the task of recognizing and extracting technical terms from scientific writing and that is highly related to NER Similar to named entities, technical terms are critical parts involved in statements. regarding outcomes and conclusions in scientific text. To identify technical terms in a scientific writing is arguably the first step to analyze the text.

The motivations for constructing a system for this task include the possibility to assist researchers in searching the vast amount of published academic papers relevant to their field. This since the analysis of academic papers would give us the possibility to construct more sophisticated academic paper search system, matching exact or similar content instead of mere keywords.

Moreover, advanced IE would allow indexing the meaning of the texts. Having a computer analyze and process academic papers might allow automatic or computer-aided inference of the knowledge contained in the papers. Especially in biomedical field, the amount of published papers in the field grows substantially in recent years [HC06] and it is impossible for a researcher to comprehend all of the knowledge. Thus, IE could provide tools to automatically extract scientific facts which would result in an improvement of the understanding of diseases and in the end better medicines.

1.3 Our method

There are several approaches to NER that can also be applied to Technical Term Extraction. However, supervised approaches which are dominant in NER are difficult to apply because the lack of annotated data. We thus examine the effectiveness of incorporating bilingual lexical resources as extra linguistic resources in Technical Term Extraction for general scientific domain. As result of our experiments,

Contact: Panot Chaimongkol, The University of Tokyo, 7-3-1 Hongo Bunkyo-ku Tokyo Japan, melsk125@nii.ac.jp

incorporation of information from bilingual lexical resources improves the recall of keywords in the datasets and new compound technical terms are also discovered by the system. We found that the bilingual resources are promising to improve Technical Term Extraction.

Section 2. of this paper lists several important works on NER and Technical Term Extraction related to our method. We provide the details of our method in section 3., including the outline of our method, problem formulation, and how we incorporate bilingual information as sets of features. In section 4., we describe how we design and conduct experiments, including resources, tools, and evaluation methods. Results of the experiments are provided in section 5., conclusions and discussions for further improvement are in section 6..

2. Related works

Several approaches have been adopted in NER and Technical Term Extraction. They include supervised and unsupervised learning methods and non-machine learning approaches.

A probabilistic model, Conditional Random Field (CRF), was proposed in [LMP01] together with a performance experiment in part-of-speech tagging. CRF model is based on an undirected graph and sequential labeling use only a special case of linear-chain. The part-of-speech tagging experiments in [LMP01] showed that CRF outperformed HMM and Maximum Entropy Markov Model (MEMM) with lower error rates but at the cost of longer training times. The superior performance of CRF to other graphical models makes it is prominently used within the field of NLP and thus became a standard.

In supervised learning algorithms, the most important key to the success of each algorithm is the choice of feature sets. But since the number of words in natural language follows Zipf's law and has long-tailed nature, most of the words are whether unseen or seen very few times in any corpus, it is difficult for the training algorithm to capture various aspects of each word. In [LW09], it is shown that extra clustered resources helps discriminative classifiers such as CRF to perform better in NER due to more informative features. The paper clustered phrases by K-Means algorithm and used the clustered information as features of token, such as the cluster itself and where the token appear in the phrase. The systems with features from phrase clustering outperformed the baseline CRF system which included only a subset of conventional features embedding contextual and shape information of the word.

The main problem of supervised learning algorithm is the difficulty to obtain annotated data. There are works dedicated to automatically generating training data. [PH11] exploited the bootstrapping method to expand the seed list to match and generate training data for supervised learning algorithms. The authors conducted experiments to extract product attribute from eBay listing titles. They compared several supervised NER algorithms such as HMM, MEMM, Support Vector Machine (SVM), and CRF. The training data for the algorithms is automatically generated from a list of known attributes. In the latter part of the paper, they grew the seed list by bootstrapping method and found that the expanded seed list made supervised algorithms to be able to detect new names, which most of them were actually spelling variants or mistakes. The system performed very well with over 90% of accuracy.

3. Method

One of the general approaches to NER is to pose the problem as sequential labeling and use machine learning methods to build the tagger system. However, large corpus in scientific domain that has technical terms annotated extensively is currently not available. It is thus necessary to utilise extra resources to provide the system with background knowledge. We focus on the bilingual lexical resources which can provide clustered information; words are clustered by their meaning and tokens are clustered by tokens in another language. It is shown in [LW09] that clustered information can help improving NER system. Bilingual lexical resources are being used by some NER systems. As far as we know, however, they are not being used in Technical Term Extraction system. In this paper, we show simple ways to incorporate the resources into Technical Term Extraction system.

3.1 Problem formulation

We pose Technical Term Extraction as sequential labeling task. Sequential labeling is the task of labeling linearly ordered data such as string of words, while assuming relations between consecutive items. We use the BIO-tagset, B tag for tokens at the beginning of a term, I tag for tokens inside a term, and O tag for tokens outside any term.

In this paper, we formulate the problem as follows:

Input A text $\mathcal{T}$, which is tokenized as $tok_1, \ldots, tok_T$

Output A length T string of BIO-tags, $tag_1, \ldots, tag_T$, where $tag_i = B$ when tok_i is the beginning of a technical term, I when tok_i is in technical term but not the beginning, and O when tok_i is not in a technical term

Example In the text "We observe that Compton-scattered photons are enhanced in a supercavity.", if only "Compton-scattered", "photons", and "supercavity" are technical terms, then the BIO-tag for tokens in this text should be as the Table 1.

3.2 Bilingual lexical resources

Bilingual lexical resources contain lists of corresponding words in two languages. The simplest example of these resources is bilingual dictionary, in which there is a list of words in a language for each word in another language.

The relation between words in two languages is not necessarily one-to-one. A word thus can appear in many entries in a bilingual dictionary coupled with different words in another language. Such cases arise from the fact that a word can be polysemous or that a word has synonyms. A polysemous word is a word which has different meanings. For example, according to Oxford Advanced Learner's Dictionary the word *protocol* could mean "the first or original version of an agreement, especially a treaty between countries; an

Token	Tag
We	O
observe	O
that	O
Compton-scattered	B
photons	B
are	O
enhanced	O
in	O
a	O
supercavity	B
.	O

Table 1: An example of BIO-tag

extra part added to an agreement or treaty" (sense 2) as in *Kyoto Protocol*, or "a set of rules that control the way data is sent between computers" (sense 3) as in *Hypertext Transfer Protocol*. The former is translated into Japanese as 議定書, while the latter as プロトコル. On the other hand, a word could have synonyms, words with the same meaning. For example, the *Krebs cycle*, a series of chemical reactions found in all aerobic organisms to produce energy, is also known as the *citric acid cycle* or the *tricarboxylic acid cycle (TCA cycle)*, while in Japanese, it is known as クレブス回路, クエン酸回路, or TCA 回路.

Natural languages are rich in variability and ambiguities. However, bilingual lexical resources can be used to resolve these ambiguities if there are multiple entries including the words. Words could be clustered by the same word in another language. For instances, the words *Krebs cycle*, *citric acid cycle*, and *tricarboxylic acid cycle (TCA cycle)* could be clustered by the word クレブス回路 in Japanese. Moreover, we can also cluster words into larger clusters by separate the words in another language into smaller subunit. This is easy for clustering with Japanese because in contrast to alphabets, Kanji characters in Japanese have meaning in themselves.

We use this clustering information as feature of tokens. Suppose the bilingual dictionary includes entries in Table 2. Bilingual information features from these entries are shown in Table 3. In the example, with Kanji character features, all four English tokens are included in the cluster 訳, while with Japanese term features, there is no cluster that includes all four English tokens.

English	Japanese
...	...
translation	並進
translation	翻訳
machine translation	機械翻訳
mechanical translation	機械翻訳
parallel translation	対訳
...	...

Table 2: Example of bilingual dictionary

English token	Japanese word feature	Kanji feature
...	...	...
machine	機械翻訳	機 械 翻 訳
mechanical	機械翻訳	機 械 翻 訳
parallel	対訳	対 訳
translation	並進 翻訳 機械 翻訳 対訳	並 進 翻 訳 機 械 対
...	...	...

Table 3: Example of bilingual information features

4. Experiments

4.1 Resources

4.1.1 Bilingual lexicon

We use bilingual lexicons of scientific terms available for internal use in National Institute of Informatics (NII). The lexicon is comprised of 244,551 entries of corresponding scientific terms in English and Japanese.

4.1.2 Abstract and author's keyword

We use a set of abstracts their corresponding keywords as dataset in training and testing. The dataset is obtained from Scholarly and Academic Information Navigator (CiNII), a scientific paper database provided by NII. We selected only papers in the domains of Information and Media, Applied Physics, and Material Science; the total number of papers is 2,079.

We randomly hold out 200 entries to manually annotate and randomly pull 40 out of 200 as development set. The remaining 1,879 entries are used as training set. Since we don't have large gold data, we tag occurrences of author's keywords in abstract to generate training set. Note that here we assume that all keywords are technical terms, while we cannot say that all technical terms are keywords. The faulty training data thus introduces noise since some technical terms will be tagged as non-terms.

4.2 Features

In the experiments, we use 3 feature sets: baseline features which contain information about the context and shape of each token, Japanese term features, and Kanji character features.

4.2.1 Baseline features

Baseline features include context features and shape features. Context features are vectors of words and parts-of-speech (POS) in 5-token window with special features for the begin and end of sentence. For token tok_i and its POS pos_i, context features are the following:

- $tok_{i-2}, tok_{i-1}, tok_i, tok_{i+1}, tok_{i+2}$,
- $tok_{i-1}tok_i, tok_itok_{i+1}$,
- $pos_{i-2}, pos_{i-1}, pos_i, pos_{i+1}, pos_{i+2}$,
- $pos_{i-2}pos_{i-1}, pos_{i-1}pos_i, pos_ipos_{i+1}, pos_{i+1}pos_{i+2}$,
- $pos_{i-2}pos_{i-1}pos_i, pos_{i-1}pos_ipos_{i+1}, pos_ipos_{i+1}pos_{i+2}$

In order to capture many aspects of the shape of each words, we adopt the set of basic shape features as described in [SPT11]. The features are defined as in Table 4, while the date regular expression is given as follows.

```
^(19|20)\d\d[- /.](0[1-9]|1[012])[- /.]\
(0[1-9]|[12][0-9]|3[01])$
```

Feature	Type	Input	Value(s)
Text	Text	Computer	Computer
Lower-cased	Text	NLP	nlp
Prefixes: sizes 3 to 5	Text	language	lan, lang, langu
Suffixes: sizes 3 to 5	Text	language	age, uage, guage
Stem [Por97]	Text	effective	effect
Is a pair of digits	Bool	12	True
Is four digits	Bool	2012	True
Letters and digits	Bool	SK125	True
Digits and hyphens	Bool	7-11	True
Digits and slashes	Bool	24/7	True
Digits and colons	Bool	3,000	True
Digits and dots	Bool	2.718	True
Upper-case and dots	Bool	H.P.	True
Initial upper-case	Bool	John	True
Only upper-case	Bool	ACL	True
Only lower-case	Bool	grep	True
Only digits	Bool	15089	True
Only non-alpha-num	Bool	%&!	True
Contains upper-case	Bool	eXternal	True
Contains lower-case	Bool	BioNLP	True
Contains digits	Bool	Y2K	True
Contains non-alpha-num	Bool	100%	True
Date regular expression	Bool	2012-01-01	True
Pattern	Text	3-26abC	0-00aaA
Collapsed Pattern	Text	3-26abC	0-0aA

Table 4: Shape features (adapted from [SPT11])

4.2.2 Bilingual information features

Bilingual information, namely Japanese term features and Kanji character features, is incorporated as explained in Section 3.2. However, function words such as *of* or *and* appear very often in the text and also in entries of bilingual lexicon. This method of would make many Japanese words to be included into these words which are mostly not parts of technical terms. We thus restricted the incorporation only for tokens which are part-of-speech tagged "ADJ" (Adjective), "ADV" (Adverb), "FW" (Foreign Word), "N" (Noun), "NP" (Proper Noun), "NUM" (Number), "VG" (Verb Gerund - Present Participle), and "VN" (Past Participle). This restriction is adopted to include bilingual information features only into noun phrases which technical terms are most likely to be.

4.3 Models

To compare the effectiveness of the two bilingual information features, we create 3 models with and without each bilingual information features. We call baseline feature set B, Japanese term feature set J, and Kanji character feature set K. The 3 models are models with B, $B+J$, and $B+K$.

4.4 Data processing

We use Natural Language Toolkit (NLTK) [BKL09], a collection of NLP tools implemented in Pyhton, in many parts of preprocessing. To split text into sentences, we use the Punkt Sentence Tokenizer included in NLTK, of which the algorithm is described in [KS06]. To split sentences into tokens, we use the Treebank Word Tokenizer in NLTK. To do the part-of-speech tagging, we use the recommended part-of-speech tagger from NLTK, in which the Maximum Entropy model is trained with Penn Treebank tagset [R$^+$96]. Our Technical Term Extraction model is based on CRF, and we use CRFsuite [Oka07] implementation of CRF learner and tagger. To perform manual annotation and to visualize the tagged result, we use an annotation tool *brat* [SPT$^+$12].

4.5 Evaluation

In both experiments, we evaluate tagging results with numerical scores: Precision, Recall, and F_1 measures. These scores are calculated by the ratio of correct tag output from the model. The number of correct tag output are counted by soft match where an instance is counted if there is any overlap with gold data and hard match where an instance is counted only if it has exactly the same span as gold data.

5. Results

The number of Technical Term spans in gold data and tagged results by each model are provided in Table 5. We see the number of spans in the tagged results all lower than the number in gold data but models with bilingual information features score higher than the one without.

Model	Number
Gold	2,646
B	1,904
B+J	1,927
B+K	1,940

Table 5: Number of spans in gold data and each model

Precision, Recall, and F_1 scores for soft match are given in Table 6 and those for hard match in Table 7. The Precision score for soft match is lower in models $B+J$ and $B+K$. On the other hand, Recall and F_1 scores are higher. We also note that for soft matching, model $B+K$ scores better than model with $B+J$. For hard matching criteria, despite the very low scores at around 20%, models $B+J$ and $B+K$ score better than model B.

Model	Precision	Recall	F_1
B	74.52%	52.72%	61.76%
B+J	74.36%	53.06%	61.93%
	(-0.16%)	(+0.34%)	(+0.17%)
B+K	74.43%	53.63%	62.34%
	(-0.09%)	(+0.91%)	(+0.58%)

Table 6: Precision, Recall, and F_1 scores for soft match

Apart from the numerical scores, we also investigate the words tagged by models even though they are not found in the keyword list of training data. These words are either not found or tagged differently in training dataset. We call

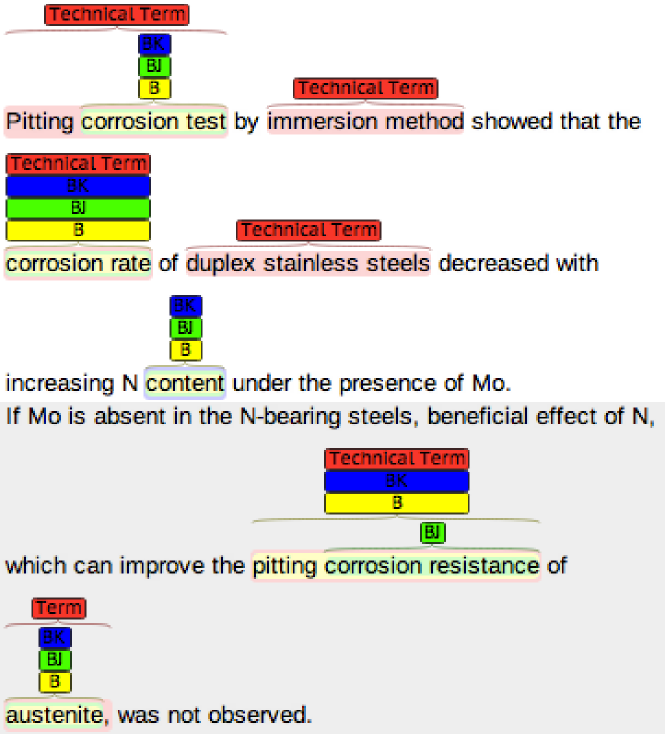
Model	Precision	Recall	F_1
B	23.37%	16.82%	19.56%
B+J	24.29%	17.69%	20.47%
	(+0.92%)	(+0.87%)	(+0.91%)
B+K	24.07%	17.65%	20.37%
	(+0.70%)	(+0.83%)	(+0.81%)

Table 7: Precision, Recall, and F_1 scores for hard match

these words out of vocabulary or OOV. We calculate the ratio of correctly tagged OOV by models. For soft matching, the ratio of model $B + J$ is lower but of model $B + K$ is higher when compared to model B. Both model $B + J$ and $B + K$, however, score better than B in hard matching criteria. The ratio is shown in Table 8. An example of tagged results visualised by brat is shown in Figure 1.

Model	Soft match	Hard match
B	77.78%	26.90%
B+J	77.33%	30.81%
	(-0.45%)	(+3.91%)
B+K	78.07%	27.27%
	(+0.29%)	(+0.37%)

Table 8: Ratio of correctly tagged OOV by models



Pitting corrosion test by immersion method showed that the corrosion rate of duplex stainless steels decreased with increasing N content under the presence of Mo.
If Mo is absent in the N-bearing steels, beneficial effect of N, which can improve the pitting corrosion resistance of austenite, was not observed.

Figure 1: Example of tagged results

6. Conclusions and discussions

We present simple ways to incorporate information from bilingual lexical resources as features into Technical Term Extraction system. Despite lower scores in Precision, models incorporated with the proposed feature set gain higher Recall score; we find this result promising. Higher Recall score implies that the bilingual information features provide more information about technical terms, allowing models to discover more technical terms as demonstrated by the OOV ratio in previous section. Even though the Precision score drops, we might be able to use the words tagged by our model as candidates for technical term and perform further analysis.

In this paper, we used English-Japanese bilingual lexicon. Our Kanji character features exploit the fact that these characters also contain meaning. In training data, the number of different Japanese term features is 106,959, while the number of different Kanji character features is only 2,301, yet both models perform as good as each other. This kind of features are much more difficult to extract in languages that use alphabets. To make use of bilingual lexicon with such languages, we would need such system that breaks down a word into smaller meaningful parts.

There are other resources that provide semantic relation such WordNet [Mil95]. These resources could also be used as a source of clustered information based on word meaning.

There are many limitations on our works, especially in experiments. In experiments, we use default settings of CRFsuite to train the models with no any parameter tuning. This might results in overfitting the training data and worse performance. Moreover, part-of-speech tagger and word tokenizer used in preprocessing are systems designed for texts in general domain. The difference in style and vocabulary, especially mathematical formulae and symbols, might lead to incorrect results. Furthermore, our training data is generated automatically and is not gold data. The model would register technical terms which are not in the keyword list as non-terms, introducing noise. Assuming that all keywords are technical terms, we could also introduce a list of non-terms and mark all other words as unknown. Doing so, it might be possible to use a method capable of training with incomplete annotations such as the one in [TKO+08].

Formulating the problem as sequential labeling also impose limitations to our system. At most one term in nested or overlapping technical terms can be extracted. We could change our problem formulation to handle overlapping terms.

Given the results of the experiments and possible future works described above, we are optimistic that utilizing bilingual lexical resources can improve Technical Term Extraction.

References

[BKL09] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python.* O ’ Reilly Media Inc., 2009.

[GR11] J. Gantz and D. Reinsel. Extracting value from chaos. *IDC iView*, 2011.

[HC06] Lawrence Hunter and K. Bretonnel Cohen. Biomedical language processing: What's beyond pubmed? *Molecular Cell*, 21(5):589 – 594, 2006.

[KS06] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Comput. Linguist.*, 32:485–525, December 2006.

[LMP01] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[LW09] Dekang Lin and Xiaoyun Wu. Phrase clustering for discriminative learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1030–1038, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[Mil95] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

[Oka07] Naoaki Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs), 2007.

[PH11] Duangmanee Putthividhya and Junling Hu. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[Por97] M. F. Porter. Readings in information retrieval. chapter An algorithm for suffix stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[R+96] A. Ratnaparkhi et al. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, pages 133–142, 1996.

[SPT11] Pontus Stenetorp, Sampo Pyysalo, and Jun'ichi Tsujii. SimSem: fast approximate string matching in relation to semantic category disambiguation. In *Proceedings of BioNLP 2011 Workshop*, BioNLP '11, pages 136–145, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[SPT+12] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL 2012 (to appear)*, Avignon, France, April 2012. Association for Computational Linguistics.

[TKO+08] Yuta Tsuboi, Hisashi Kashima, Hiroki Oda, Shinsuke Mori, and Yuji Matsumoto. Training conditional random fields using incomplete annotations. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 897–904, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.