End-to-End System for Bacteria Habitat Extraction

Farrokh Mehryary[1,2]*, Kai Hakala[1,2]*, Suwisa Kaewphan[1,2,3]*,
Jari Björne[1], Tapio Salakoski[1,3] and Filip Ginter[1]
1. Turku NLP Group, Department of FT, University of Turku, Finland
2. The University of Turku Graduate School (UTUGS), University of Turku, Finland
3. Turku Centre for Computer Science (TUCS), Finland
firstname.lastname@utu.fi

Abstract

We introduce an end-to-end system capable of named-entity detection, normalization and relation extraction for extracting information about bacteria and their habitats from biomedical literature. Our system is based on deep learning, CRF classifiers and vector space models. We train and evaluate the system on the BioNLP 2016 Shared Task Bacteria Biotope data. The official evaluation shows that the joint performance of our entity detection and relation extraction models outperforms the winning team of the Shared Task by 19pp on F-score, establishing a new top score for the task. We also achieve state-of-the-art results in the normalization task. Our system is open source and freely available at https://github.com/TurkuNLP/BHE.

1 Introduction

Knowledge about habitats of bacteria is crucial for the study of microbial communities, e.g. metagenomics, as well as for various applications such as food processing and health sciences. Although this type of information is available in the biomedical literature, comprehensive resources accumulating the knowledge do not exist (Deléger et al., 2016).

The BioNLP Bacteria Biotope (BB) Shared Tasks are organized to provide a common evaluation platform for language technology researchers interested in developing information extraction methods adapted for the detection of bacteria and their physical locations mentioned in the literature. So far three BB shared tasks have been organized, the latest in 2016 (BB3) consisting of three main subtasks: named entity recognition and categorization (BB3-cat and BB3-cat+ner), event extraction (BB3-event and BB3-event+ner) and knowledge base extraction. The NER task includes three relevant entity types: HABITAT, BACTERIA and GEOGRAPHICAL, the categorization task focuses on normalizing the mentions to established ontology concepts, although GEOGRAPHICAL entities are excluded from this task, whereas the event extraction aims at finding the relations between these entities, i.e. extracting in which locations certain bacteria live in. The knowledge base extraction task is centered upon aggregating this type of information from a large text corpus.

In this paper we revisit the BB3 subtasks of NER, categorization and event extraction, all of which are essential for building a real-world information extraction pipeline. As a result, we present a text mining pipeline which achieves state-of-the-art results for the joint evaluation of NER and event extraction as well as for the categorization task using the official BB3 shared task datasets and evaluation tools. Building such end-to-end system is important for bringing the results from the shared tasks to the actual intended users. To our best knowledge, no such system is openly available for bacteria habitat extraction.

The pipeline utilizes deep neural networks, conditional random field classifiers and vector space models to solve the various subtasks and the code is freely available at https://github.com/TurkuNLP/BHE. In the following sections we discuss our system, divided into three modules: entity recognition, categorization and event extraction. We then analyze the results and finally discuss the potential future research directions.

*These authors contributed equally.

Proceedings of the BioNLP 2017 workshop, pages 80–90,
Vancouver, Canada, August 4, 2017.

2 Method

2.1 Named entity detection

Detecting the BB3 HABITAT, BACTERIA and GEOGRAPHICAL mentions is a standard named entity recognition task, evaluated based on the correctness of the type and character offsets of the discovered text spans. In our NER pipeline, all documents are preprocessed following the approach of Hakala et al. (2016). In brief, we first convert all documents and annotation files from UTF-8 to ASCII encoding using a modified version of publicly available tool designed for parsing PubMed documents (Pyysalo et al., 2013) [1]. Next we split documents into sentences using the Genia Sentence Splitter (Sætre et al., 2007) and the sentences are subsequently tokenized and part-of-speech tagged using the tokenization and POS-tagging modules in NERsuite [2], respectively.

To detect the entity mentions we use NERsuite, a named entity recognition toolkit, as it is relatively easy to train on new corpora, yet supports adding novel user-defined features. In biomedical NER, NERsuite has been a versatile tool achieving excellent performance for various entity types (Ohta et al., 2012; Kaewphan et al., 2014, 2016), however, it is not capable of dealing with overlapping entities. Therefore, we only use the longest spans of overlapping annotated entities as our training data, ignoring embedded entities which are substrings of the longest spans.

In biomedical NER, domain knowledge such as controlled vocabularies has been crucial for achieving high performance. In this work we prepare 3 dictionaries, specific for each entity type. For BACTERIA, we compile a dictionary of names exclusively from the NCBI Taxonomy database[3] by including all names under bacteria superkingdom (NCBI taxonomy identifier 2). The *scientific names* are expanded to include abbreviations whose genus names are conventionally abbreviated with the first and/or second alphabet, whereas the rest of the names, such as species epithet and strains, remains unchanged. For HABITAT, we combine all symbols from the OntoBiotope ontology [4] and use them without any further modifications. Similar to HABITAT, we also prepare dictionary for GEOGRAPHICAL by taking all strings under the semantic type *geographical area* from UMLS database (version 2016AA) (Bodenreider, 2004). All dictionaries prepared in this step are directly provided to NERsuite through the dictionary-tagging module without any normalization. The tagging provides additional features describing whether the tokens are present in some semantic categories, such as bacteria names or geographical places. For GEOGRAPHICAL model, we also add token-level tagging results for *location* from Stanford NER (SNER) (Finkel et al., 2005) as binary values to NERsuite; 1 and 0 for location and non-location, respectively.

Although utilizing dictionary features is beneficial for NER, strict string matching tends to lead to low coverage, an issue which is also common in the categorization task. To remedy this problem, we also generate fuzzy matching features based on our categorization system (see Section 2.2) by measuring the maximum similarity of each token against the NCBI Taxonomy and OntoBiotope ontologies for BACTERIA and HABITAT respectively. Thus, instead of a binary feature denoting whether a token is present in the ontology or not, a similarity score ranging from 0 to 1 is assigned for each token. This approach is similar to (Kaewphan et al., 2014), but instead of using word embedding similarities, our fuzzy matching relies on character ngrams. We do not use these features for the GEOGRAPHICALentities, which are not categorized by our system.

In the official BB3 evaluation, NER is jointly evaluated with either categorization or event extraction system. In BB3-cat+ner task, SER (Slot Error Rate) is used as the main scoring metric, whereas in BB3-event+ner, participating teams are ranked based on F-score of extracted relations. Due to the lack of an official evaluation on NER for all entities in BB3-event+ner and for GEOGRAPHICAL in BB3-cat+ner, we use our own implementation by calculating the F-score using exact string matching criteria as our main scoring metric. In this study, we consider BB3-event+ner as our primary subtask and thus all hyper-parameters in model selection are optimized against F-score instead of SER.

2.2 Named entity categorization

In the BB3 categorization subtask each BACTERIA and HABITAT mention has to be assigned to the corresponding ontology concepts, specifically

[1] https://github.com/spyysalo/nxml2txt
[2] http://nersuite.nlplab.org/
[3] https://www.ncbi.nlm.nih.gov/taxonomy
[4] http://agroportal.lirmm.fr/ontologies/ONTOBIOTOPE

to NCBI Taxonomy and OntoBiotope identifiers respectively. This task is commonly known as named entity normalization or entity linking and various approaches ranging from Levenshtein edit distances to recurrent neural networks have been suggested as the plausible solutions (Tiftikci et al., 2016; Limsopatham and Collier, 2016).

Our categorization method is based on the common approach of TFIDF weighted sparse vector space representations (Salton and Buckley, 1988; Leaman et al., 2013; Hakala, 2015), i.e. the problem is seen as an information retrieval task where each concept name in the ontology is considered a document and the IDF weights are based on these names. Consequently, each concept name and each entity mention is represented with a TFIDF weighted vector and the concept with the highest cosine similarity is assigned for a given entity. Whereas these representations are commonly formed in a bag-of-words fashion, in our experiments using character-level ngrams resulted in better outcome. In the final system we use ngrams of length 1, 2 and 3 characters. These ngram lengths produced the highest accuracy on the official development set for both BACTERIA and HABITAT entities, each entity type evaluated separately. The TFIDF vectorization was implemented using the scikit-learn library (Pedregosa et al., 2011) and default parameter values except for using the character level ngrams instead of words.

For both included ontologies we use the preferred names as well as the listed synonyms to represent the concepts. Since the task is restricted to bacteria mentions instead of all organisms, we also narrow down the NCBI Taxonomy ontology to cover only the Bacteria superkingdom, i.e. the categorization system is not allowed to assign taxonomy identifiers which do not belong to this superkingdom. Otherwise all concepts from the used ontologies are included.

As preprocessing steps we use three main approaches: abbreviation expansion, acronym expansion and stemming. For stemming we use the Porter stemmer (Porter, 1980) and stem each token in the entities and concept names. According to our evaluation this is not beneficial for the BACTERIA entities and is thus included only for the HABITAT entities.

In biomedical literature the genus names in BACTERIA mentions are commonly shortened after the first mention, e.g. *Staphylococcus aureus* is abbreviated as *S. aureus*, but the NCBI Taxonomy ontology does not include these abbreviated forms as synonyms for the corresponding concepts. Thus, if an entity mention includes a token with a period in it, we expand the given token by finding the most common token with the same initial from all previously mentioned entities of the same type within the same document.

Another commonly used naming convention for BACTERIA mentions is forming acronyms, e.g. *lactic acid bacteria* is often referred to as *LAB*. Consequently, when we detect a BACTERIA mention with less than five characters or written in uppercase, we try to find the corresponding full form by generating acronyms for all previously mentioned BACTERIA entities by simply concatenating their initials. However, many BACTERIA acronyms do not follow this format strictly, e.g *Lactobacillus casei strain Shirota* should be shortened to *LcS* instead of *LCSS* and *Francisella tularensis Live Vaccine Strain* as *LVS* instead of *FTLVS*. Thus, instead of using strict matching to find the corresponding full form, we utilize the same character-level TFIDF representations as used for the actual categorization for these acronyms to find the most similar full form. In our evaluation, using the same approach for HABITAT entities dramatically decreased the performance hence was thus not used for this entity type (see Section 3.2).

Both of these expansion methods have similar intentions as the preprocessing steps utilized by the winning system in BB3 (BOUN) by Tiftikci et al. (2016), but our system uses more relaxed criteria for finding the full forms and should thus result in better recall at the expense of precision.

2.3 Event extraction

The BB3-event and BB3-event+ner tasks demand extraction of undirected binary associations of two named entities: a BACTERIA entity and either a HABITAT or a GEOGRAPHICAL entity; and these relations represent the locations in which bacteria live. We thus formulate this task as a binary classification task and assign the label *positive* if such relation holds for a given entity pair and *negative* otherwise.

To address this task, we present a deep learning-based relation extraction system that generates features along the *shortest dependency path (SDP)*

	Train	Devel	Test
Total sentences	527	319	508
Sentences w/examples	158	117	158
Sentences w/o examples	369	202	350
Total examples	524	506	534
Positive examples	251	177	-
Negative examples	273	329	-

Table 1: BB3-event data statistics.

which connects the two candidate entities in the syntactic parse graph. Many successful relation extraction systems have been built utilizing SDP (Cai et al., 2016; Mehryary et al., 2016; Xu et al., 2015; Björne and Salakoski, 2013; Björne et al., 2012; Bunescu and Mooney, 2005) since it is known to contain most of the relevant words for expressing the relation between the two entities while excluding less relevant and uninformative words. Since this approach focuses on a *single* sentence parse graph at a time, it is unable to detect plausible cross-sentence relations, i.e, the cases in which the two candidate entities belong to different sentences. As discussed by Kim et al. (2011), detecting such relations is a major challenge for relation extraction systems. We simply exclude any cross-sentence relations from training, development and test sets.[5] Table 1 summarizes the statistics of the data that is used for building our relation extraction system after removing cross-sentence relations.

2.3.1 Preprocessing

For preprocessing, we use the preprocessing pipeline of the TEES system (Björne and Salakoski, 2013) which automates tokenization, part-of-speech tagging and sentence parsing. TEES runs the BLLIP parser (Charniak and Johnson, 2005) with the biomedical domain model created by McClosky (2010). The resulting phrase structure trees are then converted to dependency graphs (*nonCollapsed* variant of Stanford Dependency) using the Stanford conversion tool (version 2.0.1) (de Marneffe et al., 2006).

2.3.2 Relation extraction system architecture

The architecture of our deep learning-based relation extraction system is centered around utilizing three separate convolutional neural networks (CNN): for the sequence of *words*, the sequence of *POS tags*, and the sequence of *dependency types* (the edges of the parse graph), along the SDP connecting the two candidate entities (see Figure 1). Even though the parse graph is directed, we regard it as an undirected graph and always traverse the SDP by starting the path from the BACTERIA entity mention to the HABITAT/GEOGRAPHICAL, regardless of the order of their occurrence in the sentence. Evaluation against the development set showed that this approach leads to better generalization in comparison with simply traversing the path from the first occurring entity mention to the second (with/without considering the direction of the edges).

The structure of each CNN is similar: the words (or POS tags or dependency types) in the sequence are mapped into their corresponding vector representations using an embedding lookup layer. The resulting sequence of vectors is then forwarded into a convolutional layer which creates a convolution kernel that is applied on the layer input over a single spatial dimension to produce a tensor of outputs. These outputs are then forwarded to a max-pooling layer that gathers information from local features of the SDP. Hence, the three CNNs produce three vector representations.

Subsequently, the output vectors of the CNNs and two 1-hot-encoded entity-type vectors are concatenated. The first entity-type vector represents the type of the first occurring entity in the sentence (BACTERIA, HABITAT or GEOGRAPHICAL), and the other is used for the second one. The resulting vector is then forwarded into a fully connected hidden layer and finally, the hidden layer connects to a single-node binary classification layer.

For the word features, we use a vector space model with 200-dimensional word embeddings pre-trained by Pyysalo et al. (2013). These are fine-tuned during the training while the POS-tag and dependency type embeddings are learned from scratch after being randomly initialized.

Based on experiments on the development set, we have set the dimensionality of the POS tag embeddings to 200, and for dependency types to 300. For all convolutional layers, the number of filters has been set to 100 and the window size (filter length) to 4. Finally, dimensionality of the hidden layer has been set to 100.The *ReLU* activation function is applied on the output of the convolutional layers while we apply *sigmoid* activation to

[5] Official evaluation results on the development and test data are of course comparable to those of other systems: any cross-sentence relations in the development/test data count against our submissions as false negatives.

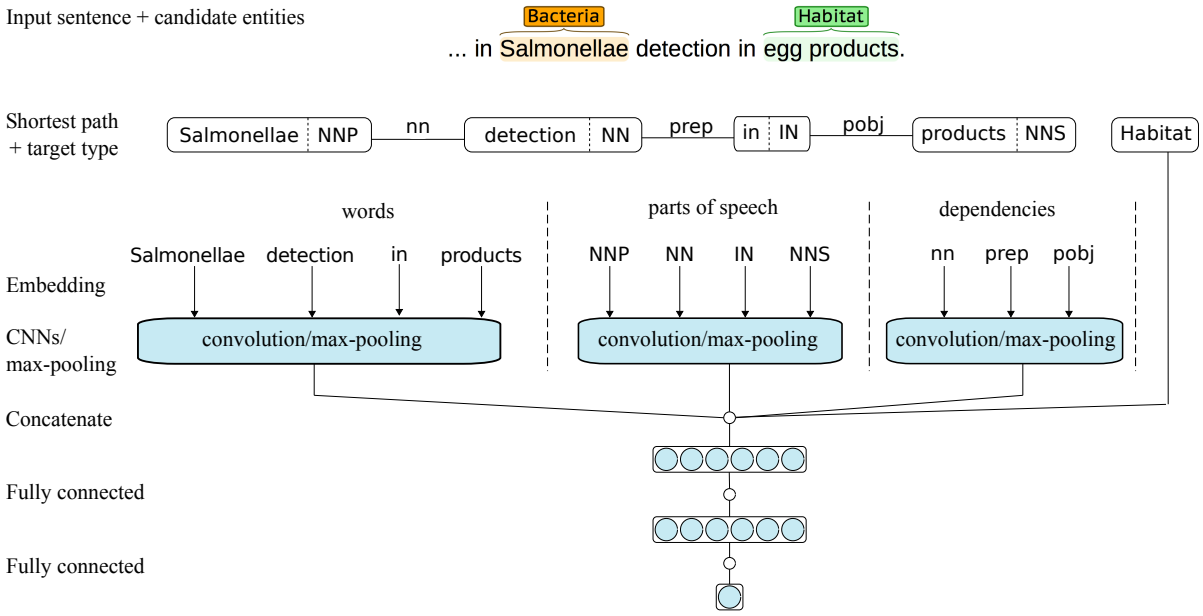

Figure 1: Proposed network architecture.

the output of the hidden layer.

2.3.3 Training and optimization

We use *binary cross-entropy* as the objective function and the *Adam* optimization algorithm (Kingma and Ba, 2014) for training the network. Applying the *dropout* (Srivastava et al., 2014) with rate of 50% on the output of the hidden layer is the only network regularization method used to avoid overfitting.

When the number of weights in a neural network is high and the training set is very small (e.g., there are only 524 examples in the BB3-event training set), the initial random state of the model can have a significant impact on the final model and its generalization performance. Mehryary et al. (2016) have reported that the F-score on the development set of BB3-event task can vary up to 9 percentage points based on the different initial random state of the network.

To overcome this problem, we implement the simple but effective strategy proposed by them, which consists of training the neural network model 15 times with different initial random states, predicting the development/test set examples and aggregating the 15 classifiers' predictions using a simple voting algorithm.

For each development/test example, the voting algorithm combines the predictions based on a given threshold parameter t: the relation is voted to be positive if at least t classifiers have predicted it to be positive, otherwise, it will be considered as a negative. Obviously, the lowest threshold value ($t = 1$) produces the highest recall and lowest precision and the highest threshold ($t = 15$) produces the highest precision and lowest recall and the aim is to find be best threshold value which maximizes the F-score.

Our experiments on the development set (using the proposed network architecture) showed that for the BB3-event task the optimal results are achieved when we train the networks for 2 epochs and set the threshold value to 4, and for the BB3-event+ner task, when we train the networks for 2 epochs and set the threshold value to 3.

3 Results and discussion

3.1 Named entity detection

For the named entity detection task, we obtain the baseline performance by training NERsuite for each entity-type independently. As shown in Table 2, the F-scores for BACTERIA, GEOGRAPHICAL and HABITAT are 0.713, 0.516 and 0.482 respectively. The baseline performance of HABITAT and GEOGRAPHICAL models is significantly lower than BACTERIA.

For all entities, adding dictionary features improves the performance of the model. A substantial improvement in F-score is found for GEOGRAPHICAL where the performance is increased

Entity/Experiment	Precision	Recall	F-score
Bacteria			
BB3	0.787	0.652	0.713
BB3 + dict	0.833	0.697	0.759
BB3 + tfidf	0.793	0.660	0.720
BB3 + tfidf + dict	0.822	0.717	0.766
BB3 + BB2 + dict	0.902	0.713	0.796
BB3 + BB1 + dict	0.893	0.721	**0.798**
Habitat			
BB3	0.589	0.407	0.482
BB3 + dict	0.649	0.465	0.541
BB3 + tfidf	0.697	0.482	0.570
BB3 + tfidf + dict	0.715	0.520	**0.602**
BB3 + BB2 + dict	0.560	0.500	0.529
Geographical			
BB3	0.667	0.421	0.516
BB3 + dict	0.719	0.605	0.657
BB3 + SNER	0.694	0.658	0.676
BB3 + dict + SNER	0.788	0.684	0.732
BB3 + BB2 + dict	0.903	0.737	**0.812**

Table 2: The performance of our named entity detection system on BACTERIA, HABITAT and GEOGRAPHICAL mentions using internal evaluation system. The models are evaluated on the BB3 development data.

by more than 14pp compared to 6pp and 5pp for HABITAT and BACTERIA, respectively. Adding fuzzy matching features further improves the F-score for HABITAT by more than 12pp compared to 8pp for BACTERIA. This result shows that having both domain knowledge and relaxed matching criteria can significantly enhance the model performance.

We improve equally the baseline performance for GEOGRAPHICAL by adding features from SNER tagging. The increase in F-score, 0.657 versus 0.676, is about the same as independently adding *UMLS-geographical area* dictionary features. Further increase in F-score is achieved by combining both features, likely due to the expanded coverage of geographical names.

The BB3 corpus is relatively small in terms of entity frequency and the number of unique entities. We explore the possibility of increasing model performance through adding additional training data from previously organized BB Shared Tasks (i.e, BB1 (Bossy et al., 2011) and BB2 (Bossy et al., 2013)). Annotations for BACTERIA mentions are available in both BB1 and BB2 Shared Tasks and we thus train NERsuite models by adding these annotations to the training data. The results show that the models, trained with additional datasets, achieve higher performance. BB1 provides a slightly better F-score than BB2, 0.798 vs 0.796.

For GEOGRAPHICAL and HABITAT entities, compatible annotations are only available from BB2 (Bossy et al., 2013), subtask 2. We thus train NERsuite for both HABITAT and GEOGRAPHICAL by using combined BB3 and BB2 data. The result for GEOGRAPHICAL is similar to the one observed with BACTERIA and additional data can increase the model F-score by more than 15pp. However, the result for HABITAT is different as F-score slightly drops from 0541 to 0.529. The best NER model for HABITAT thus remains unchanged.

Finally, we train our final model by combining training and development datasets and use hyperparameters obtained from the best performing system on development dataset. The official evaluation of the NER task jointly with either categorization or event extraction system is discussed in Section 3.2 and Section 3.3, respectively.

3.2 Categorization

To analyze our categorization approaches, we evaluate their performance on the official development set. During the development we used accuracy for evaluating the effects of different hyperparameters and preprocessing steps. To get comparable results to previous systems we, however, report the results in this paper using the precision scores from the official evaluation service. As the used ontologies form hierarchical structures, the official evaluation penalizes the incorrect predictions based on the distance from the gold standard annotations, whereas our internal accuracy evaluation measures exact matches. Our accuracy scores and the official evaluation seem to correlate to the level that all improvements validated using the accuracy score also improved the performance according to the official evaluation.

The performance of our system and various preprocessing steps are shown incrementally in Table 3. As a baseline system we use TFIDF bag-of-words representations without any of our preprocessing steps. By simply switching to character level representations the precision is increased by 1.3pp for HABITAT and 14.1pp for BACTERIA mentions.

Adding the abbreviation expansion step further improves precision for BACTERIA by 14.1pp, but does not influence HABITAT entities as most likely there are no abbreviated mentions in this category. The acronym expansion has a lesser, but still no-

ticeable impact and improves precision for BACTERIA by 4.9pp. However, applying this method to HABITAT entities decreases the performance by 4.5pp and is thus left out in the final system for this entity type. This is probably due to the fact that we consider all tokens with less than 5 characters to be acronyms, which seems to hold for BACTERIA mentions, but is a bad assumption for HABITAT entities. The final preprocessing step, stemming, improves the performance on HABITAT entities by mere 1.3pp, but has a negative impact on BACTERIA and is left out for this entity type in the final system.

The results on the official test set are consistently lower than on the development set for both entity types (see Table 4), suggesting that the hyperparameters selected based on the development set might have been slightly overfit on this data. However, our system is able to outperform BOUN (Tiftikci et al., 2016), the winning system from the BioNLP'16 BB3 Shared Task, by 1pp, 1.5pp and 1.2pp on HABITAT, BACTERIA and all entities respectively.

Since the BB3 tasks do not evaluate named entity recognition independently, but only in conjunction with either categorization or event extraction, we also report the official numbers for the BB-cat+ner task in Table 5. In this combined evaluation our system is not able to reach the performance level of the state-of-the-art system TagIt (Cook et al., 2016), but does outperform the other systems which participated in the given subtask.

Our combined system is also performing clearly worse on the test set than on the development set. Unfortunately, due to the test set being blinded, we are unable to specify the exact cause for this. However, the official evaluation service does provide relaxed evaluation modes where e.g. entity boundaries are ignored, i.e. the evaluation focuses on the categorization task. Based on these evaluations our categorization system seems to perform on the same level on both development and test sets, but the performance of our NER model drops, especially for the BACTERIA mentions. This might be simply due to overfitting on the development set, but requires further investigation.

	Habitat	Bacteria	Overall
BOW TFIDF	0.634	0.531	0.568
Char TFIDF	0.647	0.672	0.656
+ abbreviations	0.647	0.813	0.705
+ acronyms	0.602	**0.862**	0.693
+ stemming	**0.660**	0.858	0.729
Final system	**0.660**	**0.862**	**0.731**

Table 3: Evaluation of our categorization system with different preprocessing steps compared to a baseline system with TFIDF weighted bag-of-words (unigrams) representations. The scoring is produced by the official evaluation service. Any added processing step, which decreases the performance is left out for the given entity type for the following experiments.

	Habitat	Bacteria	Overall
Our system	**0.630**	**0.816**	**0.691**
BOUN	0.620	0.801	0.679

Table 4: Comparison of our entity categorization system and the best performing system in BioNLP'16 BB3 Shared Task on the test set using the official evaluation service.

	Habitat	Bacteria	Overall
	Development set		
Our system	0.645	0.377	0.553
TagIt	**0.511**	**0.303**	**0.439**
	Test set		
Our system	0.804	0.706	0.766
TagIt	**0.775**	**0.399**	**0.628**

Table 5: Official results for the combined evaluation of named entity recognition and categorization compared against the state-of-the-art system. The results are evaluated in slot error rate (SER), i.e. a smaller value is better. The scores for the TagIt system are as reported in their paper.

3.3 Event extraction

As discussed earlier, there are two tasks in the BB3 which involve extracting the relations between BACTERIA and HABITAT/GEOGRAPHICAL entities: (1) The BB3-event task, for which all manually annotated entities are given (even for the test set). This task aims to assess the performance of relation extraction systems; (2) The BB3-event+ner task, for which, entities for the test set are hidden and the aim is assessing the joint performance of the NER and the relation extraction systems.

It should be highlighted that the performance of the NER system has a direct impact on the relation extraction system and subsequently on the performance of an end-to-end system for the

BB3-event+ner task. On one hand, if the NER system produces extremely low recall outputs, the relation extraction system will fail to extract some of the valid relations, simply because it only investigates the existence of possible relations among the *given* entities. On the other hand, if the NER system provides high recall but very low precision predictions, this means that words mistakenly detected as valid entities are given to the relation extraction system. For each given entity, the relation extraction system pairs it with other provided entities in the sentence and tries to classify all candidate pairs. Hence, invalid entities will lead to generation of candidate pairs in which one or even both of the entities are actually invalid. Since the relation extraction system is trained on valid entity pairs, i.e., (BACTERIA,HABITAT) or (BACTERIA,GEOGRAPHICAL), it can easily produce a plethora of false-positives and hence, its precision will dramatically drop.

To summarize, if the NER system performance is low (low precision and/or low recall), even a very high-performance relation extraction system will not be able to compensate. Thus, when building an end-to-end system, the joint performance of NER and relation extraction should be assessed since individual performances do not reflect how efficiently the system will work in real-world applications.

The official performance of our relation extraction system alone when evaluated against the test set of the BB3-event task is 0.512 measured in F-score (0.444 recall and 0.605 precision), achieving the third place among Shared Task participants for this task.

Dataset	Overall	Habitat	Geography
Development set			
With sub-optimal entities	0.423	0.390	0.576
With optimal entities	0.429	0.395	0.604
Test set			
With sub-optimal entities	0.372	0.388	0.207
With optimal entities	0.381	0.386	0.319

Table 6: Combined performance of our named entity recognition and event extraction systems on the event+ner task reported in F-score as measured by the official evaluation service.

For the BB3-event+ner task, the official results on the development and the test set are given in Table 6. As discussed earlier, to increase the performance of the NER system, we combine the BB3 with older BB datasets. This leads to the best prediction performance (denoted as *optimal*). Thus, we report and compare the overall performance of the end-to-end system when we use these entities. To establish a fair comparison with previously published systems we also report results for models trained only on the BB3 (denoted as *sub-optimal*). As Table 6 shows, using previous BB-ST data for training the NER leads to 3pp increase in F-score of (BACTERIA,GEOGRAPHICAL) relations on the development set and about 11pp for the test set, probably due to the drastically increased performance for GEOGRAPHICAL entity detection. Unfortunately, since there are much less (BACTERIA,GEOGRAPHICAL) relations than (BACTERIA,HABITAT) relations in the data, our approach increases the overall F-score only by 1pp for the test set.

Table 7 compares the performance of our end-to-end system with the winning team in the BB3-event+ner task (LIMSI, developed by Grouin (2016)). As it can be seen in the table, our system outperforms the winning team by 19pp in F-score, achieving the new state-of-the-art score for the task. Even if we solely rely on BB3 data for the NER system, the improvement is 18pp in F-score. We emphasize that no other data than BB3 is used for training/optimization of our relation extraction system in any way.

Teams	F-score	Recall	Precision	SER
LIMSI	0.192	0.191	0.193	1.558
Our system	**0.381**	**0.292**	**0.548**	**0.891**

Table 7: Official evaluation results for BB3-event+ner test data of our system compared to LIMSI, the winning team in the Shared Task.

4 Conclusions and future work

In this work, we introduced an open-source end-to-end system, capable of named-entity detection/normalization and relation extraction to extract information about bacteria and their habitats from text. Our system is trained and evaluated on the BioNLP Shared Task 2016 Bacteria Biotope data.

According to the official evaluation, our entity detection and categorization system would have achieved the second place in BB3. Compared to the best performing system on cat+ner, TagIt, we

consider that our approach on NER can still be improved, especially for HABITAT entities. First, we consider employing a *post-processing* step in order to recover embedded entities which are not currently handled by NERsuite. An effective post-processing step should have a substantial impact on our NER system as the embedded entities accounted for over 10% of all HABITAT mentions.

Our categorization system outperforms the best performing system of BB3 by 1.2pp in the official evaluation, constituting the new state-of-the-art for this task. Our system also relies less on rule-based or heuristic preprocessing steps and uses the same general approach for both BACTERIA and HABITAT mentions suggesting that it will be more adaptable for new entity types.

As 9.6% of the HABITAT entities in the official training set have more than one gold standard ontology annotation whereas our current system is only assigning a single concept for each entity, one future work direction is to assess different ways of associating entities with multiple concepts. In the simplest form this could be implemented by defining a similarity threshold instead of selecting only the best matching concept.

Since the character level ngrams resulted in significantly better performance than our word level baseline, the exploration of character level neural approaches is also warranted for the categorization task and will be tested in the future.

Official evaluation shows that the joint performance of entity detection and relation extraction of our end-to-end system outperforms the winning team by 19pp on F-score, establishing a new top score for the event+ner task. In this work we did not use previous BB Shared Task data for training the relation extraction system. However, as a future work we would like to investigate the effect of utilizing previous BB Shared Task data.

As a future work, we would like to run our system on large-scale, on all PubMed abstracts and PubMed Central Open Access full articles to form a publicly available knowledge base.

We highlight that the methods discussed and used in this work are not only applicable for BB3 tasks and can be beneficial for other entity detection/normalization and relation extraction projects as well.

5 Acknowledgements

This work was supported by ATT Tieto käyttöön grant. Computational resources were provided by CSC - IT Center For Science Ltd., Espoo, Finland.

References

Jari Björne, Filip Ginter, and Tapio Salakoski. 2012. University of Turku in the BioNLP'11 Shared Task. *BMC bioinformatics* 13(11):S4.

Jari Björne and Tapio Salakoski. 2013. TEES 2.1: Automated annotation scheme learning in the BioNLP 2013 Shared Task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*. pages 16–25.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32(suppl 1):D267–D270.

Robert Bossy, Wiktoria Golik, Zorana Ratkovic, Philippe Bessières, and Claire Nédellec. 2013. BioNLP Shared Task 2013–an overview of the Bacteria Biotope task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*. pages 161–169.

Robert Bossy, Julien Jourde, Philippe Bessieres, Maarten Van De Guchte, and Claire Nédellec. 2011. BioNLP Shared Task 2011: Bacteria Biotope. In *Proceedings of the BioNLP Shared Task 2011 Workshop*. Association for Computational Linguistics, pages 56–64.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*. pages 724–731.

Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 173–180.

Helen V Cook, Evangelos Pafilis, and Lars Juhl Jensen. 2016. A dictionary-and rule-based system for identification of bacteria and habitats in text. In *Proceedings of the 4th BioNLP Shared Task 2016 Workshop*. Association for Computational Linguistics, Berlin, Germany.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*. pages 449–454.

Louise Deléger, Robert Bossy, Estelle Chaix, Mouhamadou Ba, Arnaud Ferré, Philippe Bessieres, and Claire Nédellec. 2016. Overview of the Bacteria Biotope task at BioNLP Shared Task 2016. In *Proceedings of the 4th BioNLP Shared Task Workshop. Berlin: Association for Computational Linguistics*. Association for Computational Linguistics, Berlin, Germany.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 363–370.

Cyril Grouin. 2016. Identification of mentions and relations between bacteria and biotope from PubMed abstracts. In *Proceedings of the 4th BioNLP Shared Task Workshop*. Association for Computational Linguistics, Berlin, Germany, pages 64–72.

Kai Hakala. 2015. UTU: Adapting biomedical event extraction system to disorder attribute detection. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, pages 375–379.

Kai Hakala, Suwisa Kaewphan, Tapio Salakoski, and Filip Ginter. 2016. Syntactic analyses and named entity recognition for PubMed and PubMed Central–up-to-the-minute. In *Proceedings of the 4th BioNLP Shared Task Workshop*. Association for Computational Linguistics, Berlin, Germany, pages 102–107.

Suwisa Kaewphan, Kai Hakaka, and Filip Ginter. 2014. UTU: Disease mention recognition and normalization with CRFs and vector space representations. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* pages 807–11.

Suwisa Kaewphan, Sofie Van Landeghem, Tomoko Ohta, Yves Van de Peer, Filip Ginter, and Sampo Pyysalo. 2016. Cell line name recognition in support of the identification of synthetic lethality in cancer from text. *Bioinformatics* 32(2):276–282.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2011. Extracting bio-molecular events from literature – the BioNLP'09 shared task. *Computational Intelligence* 27(4):513–540.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. 2013. DNorm: disease name normalization with pairwise learning to rank. *Bioinformatics* page btt474.

Nut Limsopatham and Nigel Collier. 2016. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1014–1023.

David McClosky. 2010. *Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing*. Ph.D. thesis.

Farrokh Mehryary, Jari Bjrne, Sampo Pyysalo, Tapio Salakoski, and Filip Ginter. 2016. Deep learning with minimal training data: TurkuNLP Entry in the BioNLP Shared Task 2016. In *Proceedings of the 4th BioNLP Shared Task Workshop*. Association for Computational Linguistics, Berlin, Germany, pages 73–81.

Tomoko Ohta, Sampo Pyysalo, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Open-domain anatomical entity mention detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*. Association for Computational Linguistics, pages 27–36.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program* 14(3):130–137.

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of the 5th International Symposium on Languages in Biology and Medicine (LBM 2013)*. pages 39–44.

Rune Sætre, Kazuhiro Yoshida, Akane Yakushiji, Yusuke Miyao, Yuichiro Matsubayashi, and Tomoko Ohta. 2007. AKANE system: protein-protein interaction pairs in BioCreAtIvE2 challenge, PPI-IPS subtask. In *Proceedings of the Second BioCreative Challenge Workshop*. pages 209–212.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24(5):513–523.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014.

Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Mert Tiftikci, Hakan Şahin, Berfu Büyüköz, Alper Yayıkçı, and Arzucan Özgür. 2016. Ontology-based categorization of bacteria and habitat entities using information retrieval techniques. In *Proceedings of the 4th BioNLP Shared Task 2016 Workshop*. Association for Computational Linguistics, Berlin, Germany.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 1785–1794.