Natural Language Engineering (2023), **29**, pp. 615–642
doi:10.1017/S1351324922000304

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Named-entity recognition in Turkish legal texts

Can Çetindağ[1,2], Berkay Yazıcıoğlu[1,2] and Aykut Koç[1,2,*]

[1]Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey and [2]National Magnetic Resonance Research Center (UMRAM), Bilkent University, Ankara, Turkey
*Corresponding author. E-mail: aykut.koc@bilkent.edu.tr

(Received 6 March 2021; revised 18 April 2022; accepted 25 April 2022; first published online 11 July 2022)

Abstract

Natural language processing (NLP) technologies and applications in legal text processing are gaining momentum. Being one of the most prominent tasks in NLP, named-entity recognition (NER) can substantiate a great convenience for NLP in law due to the variety of named entities in the legal domain and their accentuated importance in legal documents. However, domain-specific NER models in the legal domain are not well studied. We present a NER model for Turkish legal texts with a custom-made corpus as well as several NER architectures based on conditional random fields and bidirectional long-short-term memories (BiLSTMs) to address the task. We also study several combinations of different word embeddings consisting of GloVe, Morph2Vec, and neural network-based character feature extraction techniques either with BiLSTM or convolutional neural networks. We report 92.27% F1 score with a hybrid word representation of GloVe and Morph2Vec with character-level features extracted with BiLSTM. Being an agglutinative language, the morphological structure of Turkish is also considered. To the best of our knowledge, our work is the first legal domain-specific NER study in Turkish and also the first study for an agglutinative language in the legal domain. Thus, our work can also have implications beyond the Turkish language.

Keywords: NLP in law; NER; Turkish NER; Computational law; Named-entity recognition

1. Introduction

Being the art and science of justice, law is one of the most essential mechanisms of a healthy society. Its dependency to the ever-changing needs of societies forces law to have a dynamic structure due to economical and social dynamics, technology, and politics. Along with this dynamic structure, continuously growing quantity and diversity of legal cases further increase the burden on law professionals. Due to its strong reliance on written text, legal documents are cumulatively increasing every day. As a fact, with the advances in the field of natural language processing (NLP), legal text data can be used to lighten the burden of law professionals by utilizing proper machine assistance. Considering the speed and consistency of a machine for scanning substantial amounts of documents and potential applications, it is inevitable to use NLP in the legal domain (Aleven 2003; Martin *et al*. 2004; Ruger *et al*. 2004; Evans *et al*. 2008; Ashley and Brüninghaus 2009; Bach *et al*. 2013; Aletras *et al*. 2016; Katz, Bommarito, and Blackman 2017; Şulea *et al*. 2017a,b; Chalkidis, Androutsopoulos, and Michos 2017; Sangeetha *et al*. 2017; Virtucio *et al*. 2018; Chalkidis and Kampas 2019; Long *et al*. 2019; Ikram and Chakir 2019; Dale 2019; O'Sullivan and Beel 2019; Medvedeva, Vols, and Wieling 2020).

When elaborating on foreseeable NLP applications in the legal domain, unsurprisingly many of the explored possibilities had been toward the constitutional structures of litigation of respective countries as well as international judicial authorities (Aletras *et al*. 2016; Katz *et al*. 2017; Chalkidis *et al*. 2017; Soh, Lin, and Chai 2019). Exemplary subjects of interest may be identified as legal

document and contract summarizing and characterization (Galgani, Compton, and Hoffmann 2012; Chalkidis *et al*. 2017; Manor and Junyi 2019; Simonson, Broderick, and Herr 2019), text and judgment classification ranging from court judgments to general legislation (Nanda *et al*. 2017; Şulea *et al*. 2017a; Chalkidis *et al*. 2019a,b; Dale 2019), information extraction and retrieval of previous cases (Jackson *et al*. 2003; Finkel, Grenager, and Manning 2005; Sangeetha *et al*. 2017), question and answering (QA) systems in the legal domain (Kim, Xu, and Goebel 2017; Morimoto *et al*. 2017) as well as predicting superior court decisions (Martin *et al*. 2004; Aletras *et al*. 2016; Katz *et al*. 2017; Branting *et al*. 2018; Virtucio *et al*. 2018; Kowsrihawat, Vateekul, and Boonkwan 2018; Medvedeva *et al*. 2020). Since means of accomplishing such assorted tasks vary, overarching methodologies are incessantly evolving. As a result, to develop successful and widespread NLP applications in the legal domain, proposing new techniques and enhancing existing ones for different languages and law systems gradually attain importance. Recently, these efforts have been consolidated by Chalkidis *et al*. (2021) in *LexGLUE*, where several legal corpora and standardized benchmarks for the field are introduced.

Named-entity recognition (NER) is one of the most prominent tasks of information extraction by NLP, which can substantiate a great convenience for NLP in law due to the variety of named entities (NEs) in the legal domain and their accentuated importance in legal documents. The NER problem was introduced in the Sixth Message Understanding Conference (MUC-6) in 1995 by Grishman and Sundheim (1995). Many models have been proposed either customized for specific languages or with language-independent structures (Yadav and Bethard 2019). Initially, the task focused on only three types of NEs, which are person names (PER), location names (LOC), and organization names (ORG). These types can expand with several other entities such as date or numerical values. In more advanced applications, one can add domain-specific NEs such as protein names and drug names (Yadav and Bethard 2019), and sports teams, brands, movies, and products (Lim, Tan, and Selvaretnam 2019). Early solutions to general NER tasks usually rely on predefined rules or machine learning algorithms requiring feature-engineered content. As the interrelations within a sentence are highly important in sentence-based tasks such as NER, the initial concern is to capture such relations. Current state-of-the-art models on NER mostly rely on neural network-based models such as bidirectional long-short-term memories (BiLSTM) (Huang, Xu, and Yu 2015; Ma and Hovy 2016; Lample *et al*. 2016; Chiu and Nichols 2016; Yadav and Bethard 2019; Leitner, Rehm, and Moreno-Schneider 2019; Güngör, Güngör, and üsküdarli 2019; Li *et al*. 2022), which is a very powerful network, especially for encapsulating sequential relations. Certainly, NER is a task that also migrated to Turkish language following the advances in morphological analysis. As Yeniterzi, Tür, and Oflazer (2018) wrapped up, the first adaptations of Turkish NER were again mainly consisting of handcrafted rules (Küçük and Yazıcı 2009; Dalklç, Gelişli, and Diri 2010) and statistical models (Hakkani-Tür, Oflazer, and Tür 2002; Tür, Hakkani-Tür, and Oflazer 2003). Later, machine learning approaches dominated contemporary applications with conditional random fields (CRFs) (Yeniterzi 2011; Şeker and Eryiğit 2012; Küçük and Steinberger 2014) and neural network-based models (Demir and özgür 2014; Akdemir 2018; Güngör *et al*. 2019).

By using a legal domain-specific NER model, one can identify law articles related to verdicts, references to other cases, mentioned courts and for similar purposes. Having said that, success of such low-level tasks can improve the performance of higher level tasks such as verdict prediction by using transfer learning (Elnaggar, Otto, and Matthes 2018) and information extraction from previous litigation (Jackson *et al*. 2003). As NLP started to penetrate into law field, studies focusing on information extraction in the legal domain including NER models applied to legal documents have also started to emerge in English that utilize external resources such as ontologies and gazetteers (Cardellino *et al*. 2017; Chalkidis *et al*. 2017) or handcrafted rules (Dozier *et al*. 2010; Sleimi *et al*. 2018). Regarding legal domain-specific NER models without any external resources or language-dependent rules, the pioneering studies of Luz de Araujo *et al*. (2018) and Leitner *et al*. (2019) are designed to be applied to legal documents in Brazilian Portuguese and

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

German, respectively. However, this subfield is still understudied with only a few legal-domain-specific NER models where the number of studies for even one of the well-studied languages, English, is limited to only proof-of-concept (Vardhan, Surana, and Tripathy 2020). Moreover, to the best of our knowledge, there is no legal domain-specific NER study done for neither Turkish nor any agglutinative language.

In this paper, our objectives are to design, propose, and implement a NER model for Turkish legal documents and to lay the groundwork for further studies on this area with the corpus and the model we have presented. We also put a particular emphasis on the agglutinative structure of Turkish, so our results have also some indirect implications that can be of help in legal domain-specific NER applications for other agglutinative languages. We construct a domain-specific corpus, since one of the main limits in relatively low-resource languages, such as Turkish, is the lack of available corpora (Oflazer and Saraçlar 2018). Furthermore, we take into account the agglutinative structure of Turkish language and incorporate this feature to our model through Morph2Vec and character-level features extracted by convolutional neural networks (CNNs). Morph2Vec is a BiLSTM-based model that takes into account the morphological structure of a word while creating its vector representation (Üstün, Kurfal, and Can 2018). Since our analysis takes morphological information into account and presents a comprehensive study regarding its effects, it also presents implications and insights that can be extended to other agglutinative languages. To the best of our knowledge, our work is the first legal domain-specific NER study in Turkish and also the first study for an agglutinative language.

We compile a corpus for NER in Turkish law that contains domain-specific NEs: legislation (LEG), court (COU), reference (REF), and official gazette (OFF) along with the generic NEs, person (PER), location (LOC), organization (ORG), and date (DAT). For corpus construction, we have thoroughly examined the Court of Cassation cases in order to come up with the most prominent categories. As for word representations, we implement several alternatives and made comparisons. The models that we deployed are self-trained GloVe vectors (Huang *et al*. 2015; Lample *et al*. 2016; Chiu and Nichols 2016; Leitner *et al*. 2019), GloVe with character-level features extracted by CNNs (Ma and Hovy 2016; Chiu and Nichols 2016; Leitner *et al*. 2019), and GloVe with character embedding extracted by BiLSTM networks (Lample *et al*. 2016; Leitner *et al*. 2019; Güngör *et al*. 2019). On top of the models, we used a CRF layer as a sequence tagging model (Huang *et al*. 2015; Ma and Hovy 2016; Lample *et al*. 2016; Yadav and Bethard 2019; Leitner *et al*. 2019; Güngör *et al*. 2019; Li *et al*. 2020). We also experiment with Morph2Vec vectors (Üstün *et al*. 2018), both by replacing the self-trained GloVe vectors and finally by combining them, in order to represent the agglutinative structure of Turkish language in the most fitting way possible. We present experiments demonstrating that embeddings with additional extraction methods outperformed pure self-trained word representations highlighting the effect of the proposed methodologies on NER for agglutinative languages. With presented custom embeddings, Morph2Vec resulted in an overall degradation, where its combination with GloVe resulted in the highest F1 score.

The rest of the paper is organized as follows. In Section 2, we present the related work. Section 3 gives our NE categories in the legal domain and corpus details. We introduce our methodology and proposed architectures as well as the word representations in Section 4. In Section 5, we present our experimental results. We discuss our results in Section 6 and conclude in Section 7.

2. Related work

Providing a solid body of related work for NER in the legal domain requires coverage of several bodies of literature. Thus, we have presented our related work section under the following subtopics: (1) general NLP applications in legal texts, (2) general domain NER models, (3) NER

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

models developed for Turkish, and (4) NER models applied to legal documents. The former two of these subtopics are especially broad and require multiple references of previous work for a complete treatment, which is beyond the scope of our paper. Since covering a diverse previous work is required, we have to follow a representative and diluted coverage.

2.1 NLP applications in legal texts

The history of automated approaches to the legal domain dates back to the 1970s (Buchanan and Headrick 1970). In the late 1980s, practical applications of artificial intelligence-based approaches to law have started to appear (Francesconi *et al.* 2010; Bench-Capon *et al.* 2012; Casanovas *et al.* 2014).

Among the variety of NLP tasks that can be applied to legal documents, studies are commonly concentrated on predicting case outcomes (Ruger *et al.* 2004; Martin *et al.* 2004; Aletras *et al.* 2016; Şulea *et al.* 2017b; Katz *et al.* 2017; Branting *et al.* 2018; Long *et al.* 2019). The European Court of Human Rights (ECHR) decisions and the French Supreme Court cases are used in several works for case outcome prediction (Aletras *et al.* 2016; O'Sullivan and Beel 2019; Medvedeva *et al.* 2020). Studies for case outcome prediction also include Virtucio *et al.* (2018), Kowsrihawat *et al.* (2018), Mumcuoğlu *et al.* (2021), and Long *et al.* (2019) for the Higher Courts of the Phillipines, Thailand, Turkey, and China, respectively.

Other core tasks are automatic summarization (Galgani *et al.* 2012), text classification (Şulea *et al.* 2017a), and QA (Kim *et al.* 2017; Morimoto *et al.* 2017). A comprehensive survey on text summarization in the legal domain has been compiled by Kanapala, Pal, and Pamula (2019). To study the large-scale and extreme multi-label text classification problem, Chalkidis *et al.* (2019a,b) introduced a corpus for the legal domain (called EURLEX57k) that contains 57,000 case documents from EU's legislation database. Information retrieval applications on legal documents such as learning logical structures, labeling sentences, and finding legal facts by using machine learning methods are also increasingly grabbing attention (Ashley and Brüninghaus 2009; Bach *et al.* 2013; Sangeetha *et al.* 2017; Shulayeva, Siddharthan, and Wyner 2017). Another interesting task is capturing the relations between parts of legal documents (Nanda *et al.* 2017; Nguyen *et al.* 2018).

Since the *language of law* is very different than daily language, domain-specific word embeddings for the legal domain are required to deploy state-of-the-art deep learning models (Chalkidis *et al.* 2021). Chalkidis and Kampas (2019) presented the Law2Vec embeddings, which accelerate the developments in NLP for law. An important work to consolidate and accelerate the NLP applications in the legal domain is recently presented by Chalkidis *et al.* (2021). Their work is a consolidating collection of corpora and several standardized benchmarks for different NLP tasks. Recently, unwanted bias issues such as gender bias are also studied for legal word embeddings (Sevim, Şahinuç, and Koç 2022).

2.2 General domain NER models

Early versions of NER models mostly consist of handcrafted rules and heavily rely on feature engineering-based machine learning models such as hidden Markov models (Leek 1997; Freitag and McCallum 1999) and maximum entropy Markov models (MEMMs) (Borthwick and Grishman 1999; McCallum, Freitag, and Pereira 2000). Having replaced these models in terms of popularity, CRFs became the contemporary sequence tagging models considered for NER tasks (Lafferty, McCallum, and Pereira 2001). Still, these models poorly express nonlocal dependencies within sentences. The study of Finkel *et al.* (2005) tackled this problem with Gibbs sampling and reached 86.86% F1 score on CoNLL2003 corpus, which is a standardized corpus for a NER task (Sang and De Meulder 2003). Later, the study of Krishnan and Manning (2006) proposed a double-layer CRF model to overcome the same problem, and they reported an F1 score of 87.24% for all entities in the same dataset.

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Finkel and Manning (2009) propose a joint learning of parsing and NER. They experimented on OntoNotes Release 2.0, which consists of six different news corpora. Their results vary from 66.49% to 88.18% F1 scores. Among these six corpora, Tkachenko and Simanovsky (2012) improved two of them in OntoNotes (ABC and CNN) by using further feature engineering. They also reached a 91.02% F1 score in the CoNLL2003 corpus with the same model.

The first attempt to utilize LSTM units for a NER task was made in the work of Hammerton (2003). Limitations on computational power and word embedding techniques of the time prevented this study to stand out, resulting with a 60% F1 score in English. In later years, artificial neural network-based models with their language-independent structures and feature engineering boosted the efficiency of NER applications (Yadav and Bethard 2019). Collobert *et al.* (2011) proposed a CRF layer on top of CNN and reached a 89.59% F1 score.

Huang *et al.* (2015) experimented with four neural network architectures: LSTM, LSTM with a CRF layer, BiLSTM, and BiLSTM with a CRF layer. Models with BiLSTMs are effectively able to express and combine the forward and backward dependencies within a sentence, thus resulting in higher F1 scores. Among the reported F1 scores in the work of Huang *et al.* (2015), the most successful was BiLSTM with CRF model supported using "gazetteers," which is a type of external resource that contains a list of NEs. Concurrently, Chiu and Nichols (2016) proposed a BiLSTM model without a CRF layer while strengthening their embeddings with character-level word embeddings extracted by a separate CNN. The final results are the concatenated versions of word2vec word embeddings (Mikolov *et al.* 2013) and extracted character-level features. Best reported results were 90.91% (not supported by lexicons) and 91.62% (supported by lexicons) F1 scores.

Lample *et al.* (2016) proposed two models, which they refer as BiLSTM/CRF and Stack LSTM (S-LSTM). Unlike Chiu and Nichols (2016), they constructed character-level features with another BiLSTM rather than CNN. Their BiLSTM/CRF model with character-level features reached state-of-the-art results for four languages without any language-specific resources (English 90.94%, German 78.76%, Dutch 81.74%, Spanish 85.75%). Similar to the work of Lample *et al.* (2016), Ma and Hovy (2016) used CNN for character-level features as Chiu and Nichols (2016). This study presented the current state-of-the-art results in English (91.21% F1 score on CoNLL dataset) among the models that do not use any external resources. Peters *et al.* (2018) improve NER in the general domain by using ELMo-based contextualized embeddings as word representations and achieve 92.22 F1 score. Straková *et al.* (2019) extend the main task by investigating the nested NER structures. In their work, they used several architectures based on sequence-to-sequence encoders. They also enhanced their models with ELMo, BERT, and Flair, which are pretrained contextualized word embeddings. They have a reported 93.38% F1 score.

2.3 NER models developed for Turkish

Turkish is a morphologically rich language with its agglutinative structure. This property of Turkish language causes lower accuracy in purely statistical approaches in NLP tasks due to sparsity problem. Initially proposed by Oflazer (1994), morphological analysis was then applied to other Turkish information extraction models to increase performances by Tür *et al.* (2003). The model proposed in Tür *et al.* (2003) reached a 92.73% F1 score in NER task on their own corpus, which is widely used to evaluate Turkish NER models. Scores reported in this subsection are all performances obtained on this corpus. There exists some Turkish NER models that do not take morphological analysis into account such as Küçük and Yazıcı (2009); Küçük and Yazıcı (2012). These models are experimented on several domains such as news, history, child stories, and so on.

After CRF proved to be a successful tool for NER, work of Şeker and Eryiğit (2012) applied it to Turkish. Their model uses a morphological analyzer and two gazetteers, reaching a 91.94% F1 score. The work of Şeker and Eryiğit (2012) was also used as a part of the pipeline called *ITU Turkish NLP Web Service* as given in Eryiğit (2014), where an input text is thoroughly processed for several tasks including NER.

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

First deep learning-based approach to Turkish NER is the work of Kuru, Can, and Yuret (2016), where character-level representations are deployed instead of more popular word-level embeddings. Their model consists of a BiLSTM with five layers and a Viterbi decoder that is used to convert tag probabilities to tag sequences. They reached an F1 score of 91.30% without using any external resources. Akkaya and Can (2021) tackled Turkish NER problem with transfer learning and additional CRF layers reaching an entity level F1 score of 67.39% for a non-domain-specific noisy corpus.

In NER models, the usual way of constructing word representations is by concatenating word embeddings with character-level features, that is constructed using BiLSTM or CNN. Güngör *et al.* (2019) further advanced this approach by adding an additional component, "morphological embedding." They formed the morphological embeddings by feeding a separate BiLSTM by the morphological tags. This paper currently holds the state-of-the-art results with 92.93% F1 score. Moreover, this work also encapsulates the most recent NER results for four different morphologically rich languages, which are Czech (81.05% F1), Hungarian (96.11% F1), Finnish (84.34% F1), and Spanish (86.95% F1). Concurrently, Akdemir (2018) combined the same input configuration with joint learning of dependency parsing and NER, which was initially described by Finkel and Manning (2009). This model reached up to a 90.9% F1 score.

2.4 NER models applied to legal documents

There are few previous works for NER applied to legal documents. Successful handcrafted NER models and legal metadata extractors were presented due to the highly uniform structures of corpora from the legal domain (Dozier *et al.* 2010; Chalkidis *et al.* 2017; Cardellino *et al.* 2017; Sleimi *et al.* 2018). Work that built the foundations of a NER idea in the legal domain was conducted by Dozier *et al.* (2010) on the US legal system. Cardellino *et al.* (2017) used *Yet Another Great Ontology (YAGO)* and *Legal Knowledge Interchange Format (LKIF)* corpora to implement their models for NER on judgments of the ECHR. On the other hand, even though they do not presented their work as a NER model in the legal domain, Chalkidis *et al.* (2017) and Sleimi *et al.* (2018) covered NER task while performing information extraction from legal texts. These works focus on English legal texts by utilizing available external resources or handcrafted rules.

NER models in the legal domain that do not deploy any external resource nor handcrafted rules have also started to emerge (Luz de Araujo *et al.* 2018; Leitner *et al.* 2019; Vardhan *et al.* 2020). Luz de Araujo *et al.* (2018) introduced a NER corpus in the legal domain in Brazilian Portuguese and reached up to a 97.04% F1 score with a BiLSTM/CRF model in some NE categories that they defined. Similarly, Leitner *et al.* (2019) presented several deep learning models based on combinations of LSTMs and CRFs on German legal cases. Their paper is constructed upon two experiments, which can be titled as "fine-grained" (19 entity categories) and "coarse-grained" (7 entity categories). They reached 95.46% and 95.95% F1 scores for fine-grained and coarse-grained NER, respectively. For the case of English, in their proof-of-concept work, Vardhan *et al.* (2020) only analyzed the feasibility of NER task in the legal domain by modifying a ready-to-use NLP toolkit and reported 59.31% F1 score.

3. NEs in the legal domain and corpus preparation[a]

NLP in law is a recently developing subfield, thus finding corpora for specific applications is relatively harder than for common applications. NER tasks are an example for these specific applications and established corpora do not exist even for well-studied languages such as English. Even though there are publicly available Turkish corpora for classical NER tasks, this is not true for applications in the legal domain. We have compiled our corpus from a publicly available online

[a]All resources including the corpus we compiled and codes are available at https://github.com/koc-lab/turkishlegalner.

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 1. Named-entity tags in the legal domain

PER	Person
LOC	Location
ORG	Organization
DAT	Date
LEG	Legislation
COU	Court
REF	Reference
OFF	Official Gazette
O	Other

database called Legalbank[b]; where current and archived legislation, litigation, verdict, and cases of various institutions within the Turkish jurisdiction can be found.

Like almost all countries, there are several types of courts in the Turkish jurisdiction. Besides the Constitutional Court, which balances the jurisdiction and legislation, courts roughly divide into two categories: the Administrative and the Judicial Courts. Both of these two categories have a strict hierarchy within their member courts for providing a three-layered control mechanism that allows appeals to higher courts. The highest courts of these hierarchies are the Court of Cassation and the Council of State for the Judicial Courts and the Administrative Courts, respectively. The Court of Cassation and the Council of State operate as the supreme decision-makers in the hierarchy, having the authority to change or remove the verdicts of their respective lower courts. We compiled our corpus from the legal cases of the Court of Cassation since its database, which covers the main judicial cases, is quite large and varied.

In the following subsections, we will provide more detailed information for our NE categories in the legal domain, corpus construction process, and the resulting statistics.

3.1 NE categories in the legal domain

When NER has been first introduced, it has been constructed with three main NE types: PER (Person), LOC (Location), and ORG (Organization). However, for domain-specific applications, these types can be expanded to any desired number. Leitner *et al.* (2019) used NRM (Legal Norm), REG (Case-by-case regulation), RS (Court Decision), and LIT (Legal Literature) in their work on German NER in the legal domain. However, due to the disparity of legal languages across nations, we did not migrate their classes directly but created our own classes. Still, some of our classes have an irrefutable correspondence with the classes they have. In total, we have decided on eight distinct NE tags, of which four are NEs in the legal domain. These are given in Table 1, extending common NEs (PER-LOC-ORG-DAT) with the task-specific ones (LEG-COU-REF-OFF).

LEG is the abbreviation for the legislation relevant to the verdict and intuitively is the most crucial information of an arbitrary court case. COU is the abbreviation for court type. Court types are very important in law since each court has its specific authorities, areas of expertise, and features. Therefore, court types are highly relevant for information extraction from legal texts. REF is the short for reference if there is referral to a previous litigation. Extraction of references are significant in NLP applications in the legal domain, because previous litigation and court verdicts are extremely important in law. By using this information, one can find related cases easily and

[b]Legalbank database is available at https://legalbank.net/arama.

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 2. Example sentence for IOB scheme

Macron	gave	a	speech	at	the	UN	general	assembly	in	New	York	.
B-PER	O	O	O	O	O	B-ORG	O	O	O	B-LOC	I-LOC	O

their verdicts, which are likely to apply to the current case at hand as well. Finally, OFF is the short for the official gazette, and it is also critical to track the effects of new or altered legislation on cases.

3.2 Tagging scheme

There are two common tagging schemes for NER tasks, which are Inside-Outside-Beginning (IOB) (Sang and De Meulder 2003) and Inside-Outside-Beginning-End-Singleton (IOBES), adding "Singleton" and "End" to the previous one (Akdemir 2018). In our paper, we use IOB Scheme. In IOB scheme, Out (O) is used for non-NE, Begin (B) for the initialization of a NE, and Inside (I) for an internal word of a NE. An example for this scheme is given in Table 2.

3.3 Labeling guidelines

Tag Independent Guidelines

Two common steps of text preprocessing are lowering all cases and removing the punctuation. We have intentionally avoided these steps in order to preserve semantic effects of both capital letters and punctuation, since they are more important in legal text than in regular ones. It is also common to use capital letters in NER tasks in order to distinguish capitonyms. On the other hand, to extract information without any ambiguity, we have avoided punctuation removal phase since legislation are frequently referred with substantial amount of punctuation. Following is an example expression of legislation before and after removing punctuation, and the ambiguity in numeric values is irrefutable:

- 5271 sayılı CMK'nın 34/1, 230, 289/1-g ve 1412 sayılı CMUK'nın 308/7. maddeleri
- 5271 sayılı CMKnın 341 230 2891g ve 1412 sayılı CMUKnın 3087 maddeleri
- *Translation: Articles 34/1, 230, 289/1-g of the Law on Criminal Procedure, numbered 5271 and Article 308/7 of the Law on Criminal Procedure, numbered 1412*

PER Tags

- Some people are referred with their titles throughout legal documents. We have accepted such titles as PER entity.
- Due to the privacy concerns, some person names involved in the legal documents that we compiled were already replaced with *ellipsis* or represented with their initials. We have labeled these ellipsis as PER in our corpus to protect the integrity. This is also a secondary reason for avoiding punctuation removal. We did not apply any preprocessing to the given names.

LOC Tags

- In Turkish legal system, local court names contain names of their respective administrative regions. In fact, this segment of court name is particularly significant since it also provides

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

hierarchical information. We accepted this regional part as a part of the COU entity rather than LOC entity.

ORG Tags

- Organizations that are presented with their abbreviations are accepted as an ORG entity with a single word.
- Some of the organizations were originally replaced with ellipsis or presented with their initials for privacy concerns as in PER case. These are accepted as ORG entity.

DAT Tags

- There is no unified format for dates throughout the documents. Different formats of dates (*xx/xx/xxxx*, *xx. month of xxxx*) are all accepted as DAT entity.
- Dates that are included in the reference (REF) and official gazette (OFF) entities are not accepted as DAT because they are parts of the description of the entity in REF or OFF.

LEG Tags

- Some legislations are referred as "ayn Kanunun . . ." (". . . of the same law"), since they are mentioned before in the legal document. We accepted "ayn" ("of the same") as a part of the LEG entity.

COU Tags

- Since the names of administrative regions in court names are important for the hierarchical characteristics of courts, we have accepted the location names within the court names as part of the COU entity.

REF Tags

- Some REF names included dates in their content as some cases are described by their dates of verdict. These dates are included in the REF entity, not the DAT.

OFF Tags

- Since the official newspaper issues were described by dates they were published, most of them also included dates. These names are included in the OFF entity, not the DAT.

Our corpus has been labeled manually by the first author by using the guidelines given above. The guidelines are prepared with the help of an academic consultant from law school.

3.4 Corpus statistics

We have manually labeled 350 cases from the Court of Cassation. This corresponds to 2198 sentences, 5311 NEs, and 123,173 tokens in total. We have randomly divided this corpus into training and test sets with 1758 and 439 sentences, respectively. Tag statistics for each set are given in Table 3.

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 3. Named-entity distribution for the training and test sets

Tag	Training count	Test count	Total count
PER	1243	344	1587
LOC	46	5	51
ORG	241	48	289
DAT	761	189	950
LEG	1396	307	1703
COU	268	67	335
REF	245	63	308
OFF	67	21	88
O	93,738	24,124	117,862

There are four resulting documents, which are sentence–tag pairs of the training set and the test set. Total size of these documents is 1.24 MB.

4. Methodology

Legal NER differentiates from usual NER. One of the major differences is the existence and variety of specific NE types. Legal documents contain numerous references to specific NEs such as jurisdictions and courts. Standard NER methods are designed and work on general corpora (predominantly for tweets, customer reviews, news, etc.). These models are not directly suitable for processing legal documents, since they cannot detect domain-specific entities. Second is the style difference between legal language and daily language (e.g., tweets, customer reviews, and news articles) for which most standard NER models are developed. In legal language, sentences tend to be longer (sometimes up to hundreds of words) and more complex with several eccentricities such as structure, capitalization, and punctuation. Specifically, the longest sentence in our corpus has 408 tokens, and the overall average is 56.04 tokens per sentence.

To perform NER task for our proposed NEs in the legal domain, we incorporate state-of-the-art models for the general NER task with the morphological needs of the Turkish language. Our models can be differentiated from each other by the architectures they deploy and also how they utilize word embeddings and handle character-level features. Combinations of several different structures constitute the backbone of our models. In order to delineate our final proposed architectures for NER task in the legal domain better, we first present the building blocks of our final models in the following two subsections. In Section 3, we first address the architecture-wise building blocks, which are BiLSTM and CRF layers. These two components are important since they act both in the main backbone of our models and in extracting character-level features. We then present word embedding models we deploy in Section 4.2. These models are several combinations of three main approaches. Namely, we consider several combinations of trained GloVe word representations (Pennington *et al.* 2014), trained Morph2Vec embeddings (Üstün *et al.* 2018), and extracted character-level features. Finally, in Section 4.3, we present our proposed models for NER in the legal domain for Turkish.

4.1 NER model components in the legal domain

4.1.1 BiLSTM layer

Recurrent neural networks (RNNs) are a type of neural network which are widely used to process sequential data such as time series. Even though RNNs are useful for sequential data,

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

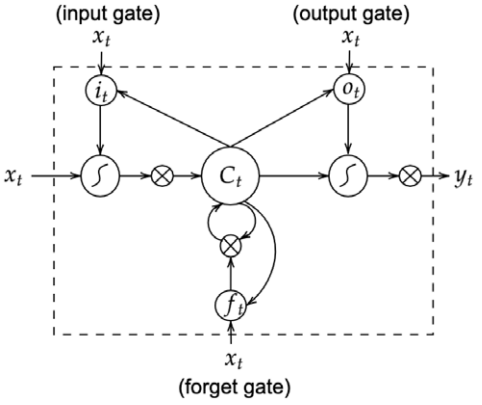

Figure 1. A single LSTM unit.

they suffer from the vanishing gradient problem when processing long sequences of data. Thus, these networks are unable to fully capture the long-term dependencies within a sequential input. Processing natural languages requires handling of long-term dependencies within sentences for coherent extraction of information. LSTM has been introduced to overcome this weakness (Hochreiter and Schmidhuber 1997).

LSTM structure captures long-term dependencies by additional gates, such as forget gate, which determines the proportion of the information that should be remembered through the process. The structure of a single LSTM unit is given in Figure 1, and the following equations reflect the mathematical background of LSTM architecture (Huang *et al.* 2015):

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{yi}y_{t-1} + W_{ci}c_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{yf}y_{t-1} + W_{cf}c_{t-1} + b_f) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{yc}y_{t-1} + b_c) \\ o_t &= \sigma(W_{xo}x_t + W_{yo}y_{t-1} + W_{co}c_t + b_o) \\ y_t &= o_t \odot \tanh(c_t), \end{aligned} \tag{1}$$

where i, f, and o are *input*, *forget*, and *output* gates, respectively. x, y, and c are *input*, *hidden*, and *cell* vectors, respectively. σ is the logistic sigmoid function. Finally, W's are weight matrices between realizations and units denoted by the two letters in the subscripts. For example, W_{xi} is the weight matrix between the realization x and unit i.

In our work, we used the BiLSTM architecture given in Graves and Schmidhuber (2005), which examines a sentence both in forward and reverse orders to catch and relate forward and backward dependencies. This is achieved by feeding a separate LSTM with reversed sequence of words. The final output is reached by concatenating forward and backward LSTM outputs.

4.1.2 CRF layer

Sequence tagging tasks are widely necessary in several applications such as part-of-speech (POS) tagging, syntactic disambiguation, and NER. Hidden Markov models (HMMs) are the earliest models that strongly rely on the conditional independency. Due to the high dependency in NER tasks (i.e., I-PER tag can only come after B-PER tag), HMMs are not the superior sequence tagging model in the said models (Lample *et al.* 2016). Non-generative models, such as MEMMs, are successful for dependent environments. However, they suffer from being biased to the most frequently detected transitions during the training (Lafferty *et al.* 2001).

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

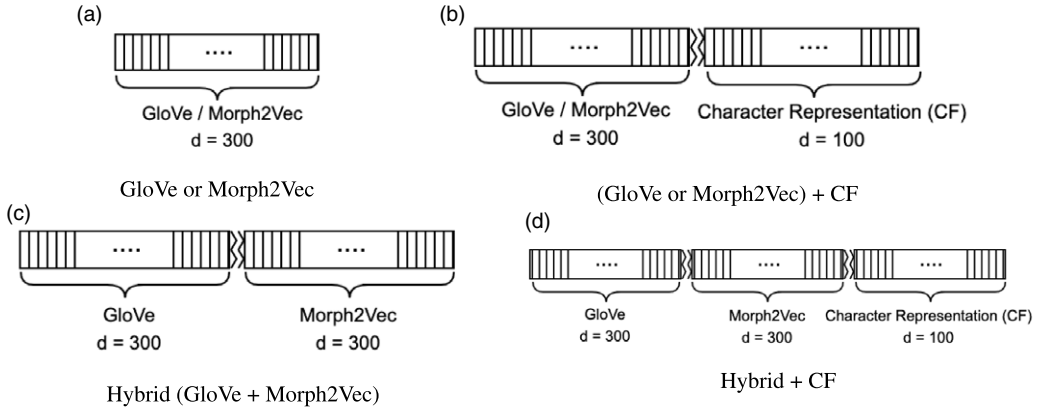

Figure 2. Word representation models.

On the other hand, since they do not require conditional independency between features to tackle the biasing problem, CRFs are proven to have a better performance in sequence tagging tasks. The main difference of CRF networks is that they have a single exponential model for the joint probability of the entire sequence of labels given the observation sequence (Lafferty *et al.* 2001). CRFs are used to extract the most probable output per element in a model via defining conditional probabilities observed on training data. Mainly, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ is the set whose elements are vertices and the set $\mathcal{E}$ holds the paired vertices, called edges. In our case, this is a linear chain of tags and corresponding tokens. Then, observation variable $\mathbf{X}$ is defined as the sequence of tokens, and a random variable $\mathbf{Y} = \mathbf{Y}_{v \in V}$ can be defined as a possible tag from the vertices $\mathcal{V}$ of $\mathcal{G}$. Resulting $(\mathbf{X}, \mathbf{Y_v})$ is a CRF that random variables $\mathbf{Y}$ are conditioned on $\mathbf{X}$, obeying the Markov property on $\mathcal{G}$:

$$P(Y|X) = \frac{exp\left(\sum_j w_j F_j(X, Y)\right)}{\sum_{Y'} exp\left(\sum_j w_j F_j(X, Y')\right)}, \tag{2}$$

where $F_j(X, Y)$ denotes the feature function and w_j denotes corresponding weights. Both $F_j(X, Y)$ and w_j are trained with back-propagation. Equation (2) can be interpreted as the inner summation of feature functions on a set of vertices normalized with summation over all features on $\mathcal{G}$. Tuning this, the output is given as the tag that has the maximum likelihood, defined on $\mathbf{Y}$ for each observation variable (token in our corpus) $\mathbf{X}$.

4.2 Word representations

As our word representations, we used six different combinations of trained GloVe embeddings, trained Morph2Vec embeddings, and extracted character representations in order to feed our models. We present these combinations of embeddings in Figure 2. Both GloVe and Morph2Vec are used separately as well as concatenated to each other as shown in Figure 2a and c. The model that uses them in the concatenated mode is called as hybrid model. These three options are considered as base word embeddings. Building on them, additional character-level features (CFs) are included as supplementary models as given in Figure 2b and d. CF is appended to base models, denoted as (GloVe/Morph2Vec + CF) and (GloVe + Morph2Vec + CF), respectively. In the following sections, we will refer (GloVe + Morph2Vec) by *Hybrid*. In what follows, we provide details on these word representation alternatives.

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

4.2.1 GloVe representations

We used publicly available source code of GloVe (Pennington *et al*. 2014) to train our base word embeddings. We trained GloVe on our custom corpus that includes 50,000 documents containing 23,305,357 tokens and 438,099 types. These 50K documents are extracted from *Lexpera*,[c] which is an online database for Turkish legal documents. Since GloVe uses unsupervised learning to construct embedding vectors, we were able to use this corpus directly without any annotation process. Dimension of these base embeddings are 300.

4.2.2 Morph2Vec representations

In morphologically rich languages such as Turkish, surface form and meaning of a word can vary remarkably from its root. Thus, for representing the linear relationships between words in these languages, the representation power of the pretrained word embeddings is relatively poor. A very intuitive way to tackle this problem is considering prefixes and suffixes while creating word embeddings. Güngör *et al*. (2019) used a two-level morphological analyzer introduced by Oflazer (1994) to analyze suffixes of a word.

Besides the hard-coded pipelines, Morph2Vec is a BiLSTM-based model presented by Üstün *et al*. (2018), which takes into account the morphological structure of a word while creating its vector representation. The idea of the proposed embeddings is to train words with its morphological segmentations. Üstün *et al*. (2018) used sub-word features to increase the Spearman correlation of embeddings from 0.483 (word2vec) to 0.529 for Turkish language. Üstün *et al*. (2018) also proposed an unsupervised segmentation model, along with the Morph2Vec model, which allows us to use their method effectively. We trained Morhp2vec with the default value of 300 for the dimension on our custom-made 50,000 document corpus. We then feed our learning models with Morph2Vec alone or in concatenation with other embeddings.

4.2.3 Character-level features

Usage of character-level features is another way to improve model performance, while working on morphologically rich languages on NLP tasks. Instead of using sub-word information, character patterns in a word can be examined with this technique.

Character-level features (CFs) are extracted with two different neural architectures, which are BiLSTM and CNN. Considering that each word is a sequence of letters, processing these sequences by using a BiLSTM architecture allows us to capture each character embedding. Similar to BiLSTMs, CNNs are again very powerful in capturing complex relations. In both techniques, each column of the final weight matrix of the trained neural architecture corresponds to a character embedding. CFs are just the concatenated versions of these character embeddings. Applications of these extracted features to substantiate NER tasks are presented by Lample *et al*. (2016) and Chiu and Nichols (2016), respectively, for BiLSTM and CNN. Extracted CFs are concatenated with GloVe, Morph2Vec, and Hybrid representations in separate experiments, where the dimension for extracted CFs is 100. We have experimented with both extraction techniques and compared the results.

To sum up, we have six different word representation alternatives as given above with three of them containing CF representations. By using two different techniques to obtain CFs, we end up with nine different configurations. We explicitly tabulate each one of them with their dimensions in Table 4, where concatenation is denoted with +.

4.3 NER models in the legal domain

NER models we constructed to address domain-specific NER for Turkish legal documents rely on three different architectures and three different base word embeddings, which are GloVe,

[c] https://www.lexpera.com.tr/.

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 4. Word representation combinations and their dimensions used in experiments

#	Word representations	Dimension
1	Trained GloVe Embeddings only	300
2	Trained GloVe Embeddings + CF extracted with BiLSTM	400
3	Trained GloVe Embeddings + CF extracted with CNN	400
4	Trained Morph2Vec Embeddings only	300
5	Trained Morph2Vec Embeddings + CF extracted with BiLSTM	400
6	Trained Morph2Vec Embeddings + CF extracted with CNN	400
7	Hybrid Word Embeddings only	600
8	Hybrid Word Embeddings + CF extracted with BiLSTM	700
9	Hybrid Word Embeddings + CF extracted with CNN	700

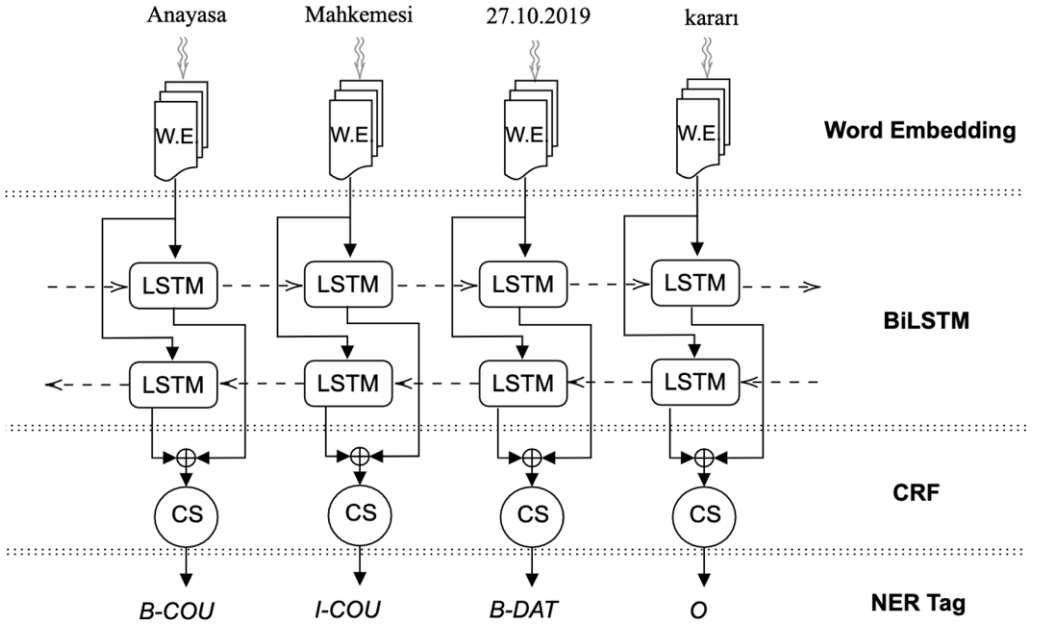

Figure 3. LSTM-CRF model. CS stands for a single unit of CRF.

Morph2Vec, and Hybrid. Our first and base architecture is the LSTM-CRF (LC for short) model (Figure 3), where character-level features are not considered explicitly (#1, #4, and #7 in Table 4). Turkish NER in the legal domain is therefore characterized in the embedding part of this network. This model is the most basic approach and generalizable for NER in other languages. In this base architecture and in the following architectures stemmed from LSTM-CRF, 1-layer BiLSTM is used.

Since in Turkish, sequential structure of suffixes in a word create diversified lemma, another LSTM unit for extracting the features within words are proposed. With this regard, we used LC model with character-level features extracted by another BiLSTM as illustrated in Figure 4, and we refer this model as LSTM-LSTM-CRF (LLC for short) model (#2, #5, and #8 in Table 4). With

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

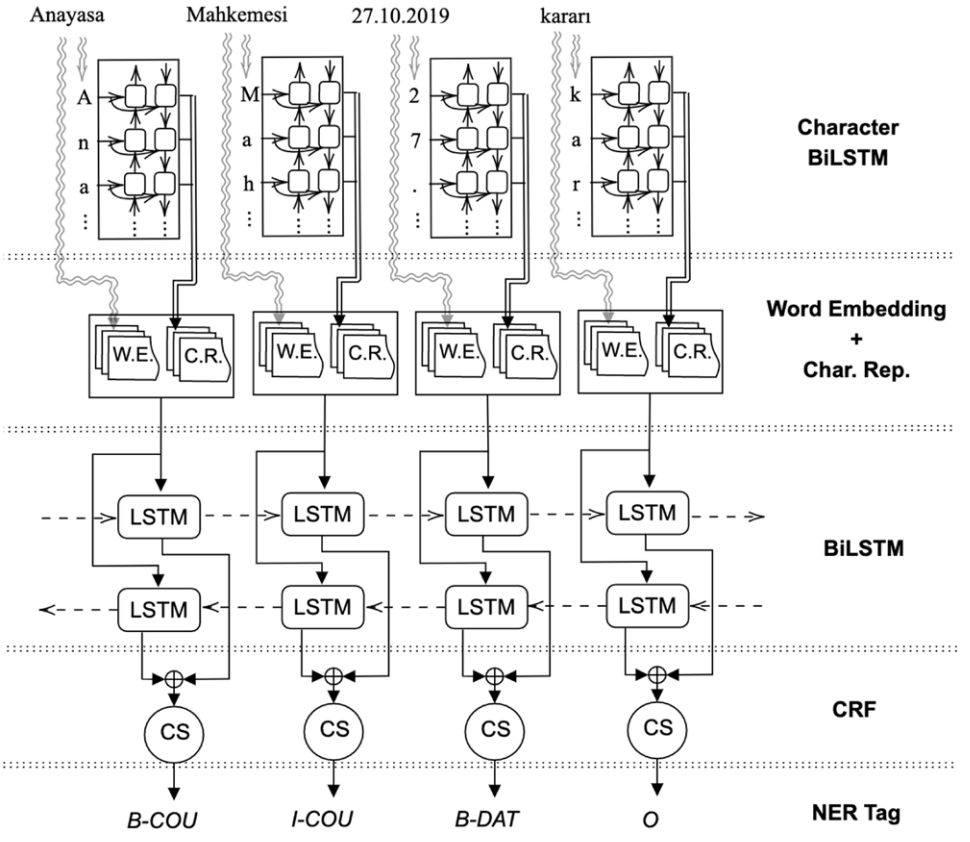

Figure 4. LLC model. CS stands for a single unit of CRF.

this model, we aim to enhance upon the performance of LC models in the domain of Turkish legal texts.

The last architecture is similar to the LSTM-LSTM-CRF model. However, in this architecture, we extracted the character-level features with CNN as demonstrated in Figure 5, and thus we refer it as CNN-LSTM-CRF (CLC for short) model (#3, #6, and #9 in Table 4). While character-level features are in a sequential structure for agglutinative languages, CNN is also a promising approach in the sense of capturing not only all suffices but also the most defining ones in terms of changes in meaning. Legal texts are in a form that is exceedingly formalized and well defined in order to prevent misinterpretations. With this structure of the input, effect of individual words and their different morphological structures are designed to be captured well enough without the need of any sequential extraction.

5. Experiments and results[d].

In this section, we have presented quantitative and qualitative results of our work on our specifically compiled and labeled legal NER corpus. Furthermore, we compare the performance of our architectures on the mostly studied general domain NER corpora both in English and Turkish. A comprehensive quantitative performance comparison between models and different word

[d] All resources including the corpus we compiled and codes are available at https://github.com/koc-lab/turkishlegalner

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

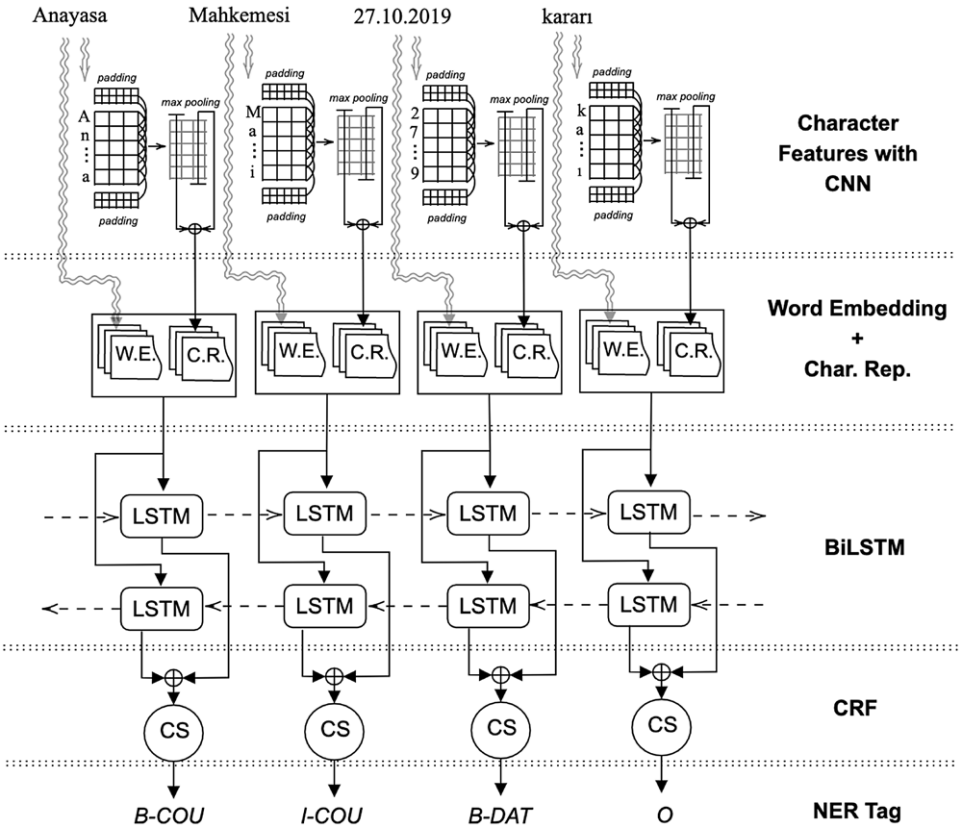

Figure 5. CLC model. CS stands for a single unit of CRF.

representations is given in the first subsection without diving into the semantic performances of different combinations. In the second subsection, we investigate the semantic performances by going through some distinctive qualitative examples. Different combinations of models and word representations are compared to address specific semantic cases. We limit our qualitative comparisons to include some of the best model–representation combinations (according to their scores in Section 5.1) instead of giving individual examples for each combination to keep results succinct. In the third subsection, we evaluate the performance of our architectures on CONLL2003 (English NER in general domain) and Tür *et al.* (2003) (Turkish NER in general domain) corpora based on the results that are presented on previous works. These comparisons are intended not to propose state-of-the-art NER models but to only show that our baseline models for legal NER in Turkish do not deviate much in general NER tasks in English and Turkish.

5.1 Quantitative results

We fed our NER models (LC, LLC, and CLC) with three different base word representations, which are GloVe, Morph2Vec, and Hybrid embeddings. Training and test procedures were done using the manually tagged corpus presented in Section 3.4. The main focus of different combinations of embeddings and models is capturing the effect of character- or morpheme-level extraction done at different levels. Extra CNN and LSTM layers enable additional character features concatenated to the initial word embedding. In addition, base word embedding may also contain morpheme-level information as it is the case for Hybrid and Morph2Vec. Therefore, we mainly

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 5. GloVe as the selected word embedding (all metrics are given in percentages.)

Named entity	LSTM-CRF			LSTM-LSTM-CRF			CNN-LSTM-CRF		
	PREC	*REC*	*F1*	*PREC*	*REC*	*F1*	*PREC*	*REC*	*F1*
PER	95.61	95.06	95.34	95.49	98.55	97.00	94.44	98.84	96.59
LOC	0.00	0.00	0.00	0.00	0.00	0.00	33.33	20.00	25.00
ORG	58.70	56.25	57.45	78.72	77.08	77.89	83.33	83.33	**83.33**
DAT	85.86	86.77	86.32	92.23	94.18	93.19	92.23	94.18	93.19
LEG	80.07	79.80	79.93	89.90	89.90	89.90	87.58	89.58	88.57
REF	82.14	73.02	77.31	88.33	84.13	86.18	90.00	85.71	**87.80**
COU	85.92	91.04	88.41	90.00	94.03	91.97	90.00	94.03	91.97
OFF	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Total	86.25	85.34	85.80	91.75	92.62	92.18	90.93	93.10	92.00

structured our experiments to highlight the differences between the effects of including multiple representations. Furthermore, effects of obtaining them with CNN or LSTM in between the structures of our proposed models are investigated with respect to their class based and overall scores.

Results are clustered and presented in a similar manner, where each table refers to the results of a respective base word representation for proposed models. They are evaluated with F1 score metric due to the highly unbalanced distribution of tags (e.g., O class: 24,124; LOC class: 5). This is also the reason why we do not consider individual entity-based accuracy scores but evaluate individual F1 scores. Tables 5, 6, and 7 present results in terms of Precision, Recall, and F1 score for each base word embedding GloVe, Morph2Vec, and Hybrid, respectively. The best results for F1 score within each word representation (for all Tables 5, 6, and 7) are given underlined, separately for each NE as well as total score. The best results for a class among all the inter-model results are also given in bold across Tables 5, 6, and 7.

To summarize and compare all models with their best-performing word representation variant, Table 8 reiterates the best results of each model and embedding with respect to their overall F1 performance as well as accuracy. The main interest is F1 score, since we deal with highly unbalanced sets.

Along with the metric-wise results presented in Tables 5, 6, 7, and 8, we also provide count statistics for each model in Table 9. First two columns are the properties of our corpus, so they are equal for all models. This table mainly extricates the number of recognized and correctly recognized entities in the test set. Comparing Tables 8 and 9, we verify that selection of F1 score as the main metric is validated in the sense that F1 results are more consistent.

To substantiate our results, we extend our experiments using k-fold cross-validation with all models. To obtain a controlled comparison environment, the same random order is used for all models. We used cross-validation to evaluate the performance of our models with respect to randomized partitions. The results are given in Figure 6, where the quantile distributions, maximum and minimum points, the median, and outlying points (if exist) of the test runs are plotted. Due to the limited size of our corpus, we started with only five partitions as given in Figure 6a to prevent insufficient test sizes. We encoded the model names on the lateral axis using their initials, where one can observe LC models perform poorly compared to the others. Among LLC and CLC results, a comparable performance is observed where Morph2Vec approach performs slightly

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 6. Morph2Vec as the selected word embedding (all metrics are given in percentages.)

Named entity	LSTM-CRF			LSTM-LSTM-CRF			CNN-LSTM-CRF		
	PREC	*REC*	*F1*	*PREC*	*REC*	*F1*	*PREC*	*REC*	*F1*
PER	94.40	93.02	93.70	96.23	96.51	96.37	95.18	97.67	96.41
LOC	0.00	0.00	0.00	100.00	20.00	**33.33**	0.00	0.00	0.00
ORG	63.64	29.17	40.00	66.00	68.75	67.35	79.55	72.92	76.09
DAT	91.98	78.84	84.90	87.06	92.59	89.74	91.62	92.59	92.11
LEG	76.82	75.57	76.19	85.85	88.93	87.36	87.90	89.90	88.89
REF	75.44	68.75	71.67	86.67	82.54	84.55	90.00	85.71	**87.80**
COU	86.15	83.58	84.85	88.89	95.52	92.09	87.14	91.04	89.05
OFF	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Total	86.08	79.98	82.92	89.04	91.09	90.06	90.89	91.76	91.33

Table 7. Hybrid word embeddings as the selected word embeddings (all metrics are given in percentages.)

Named entity	LSTM-CRF			LSTM-LSTM-CRF			CNN-LSTM-CRF		
	PREC	*REC*	*F1*	*PREC*	*REC*	*F1*	*PREC*	*REC*	*F1*
PER	95.16	97.09	96.12	95.75	98.26	96.99	95.77	98.84	**97.28**
LOC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ORG	68.89	64.58	66.67	76.09	72.92	74.47	80.00	83.33	81.63
DAT	86.93	91.53	89.18	91.79	94.71	**93.23**	88.32	92.06	90.16
LEG	80.06	82.41	81.22	90.61	91.21	**90.91**	89.10	90.55	89.82
REF	86.44	80.95	83.61	88.33	84.13	86.18	89.83	84.13	86.89
COU	89.86	92.54	91.18	89.71	91.04	90.37	91.30	94.03	**92.65**
OFF	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Total	87.26	88.60	87.93	91.92	92.62	**92.27**	91.16	92.82	91.98

Table 8. Overall best results for each model and word embedding. All metrics are given in percentages

	LSTM-CRF		LSTM-LSTM-CRF		CNN-LSTM-CRF	
	F1	*ACC*	*F1*	*ACC*	*F1*	*ACC*
GloVe	85.80	98.39	92.18	99.03	92.00	98.86
Morph2Vec	82.92	97.60	90.06	98.87	91.33	98.94
Hybrid	87.93	98.45	92.27	99.05	91.98	99.03

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 9. General entity count statistics of the test set

Model	# of Tokens	# of Entities	# of Found Entities	# of Correct Entities
LSTM-CRF (GloVe)			1033	891
LSTM-CRF (Morph2Vec)			970	835
LSTM-CRF (Hybrid)			1060	925
LSTM-LSTM-CRF (GloVe)			1054	967
LSTM-LSTM-CRF (Morph2Vec)	25,168	1044	1068	951
LSTM-LSTM-CRF (Hybrid)			1052	967
CNN-LSTM-CRF (GloVe)			1069	972
CNN-LSTM-CRF (Morph2Vec)			1054	958
CNN-LSTM-CRF (Hybrid)			1063	969

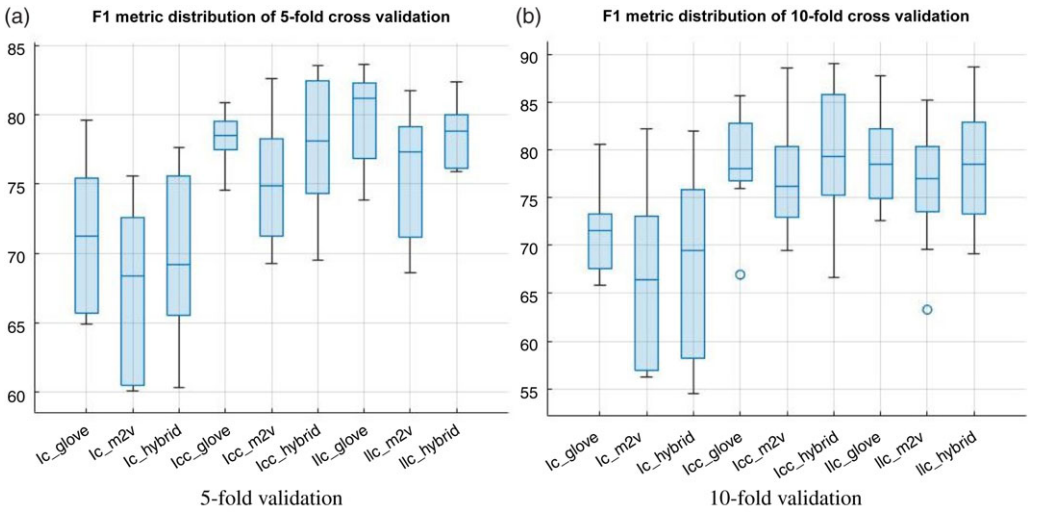

Figure 6. K-fold cross validation results.

worse. Extending our validation to 10 partitions on Figure 6a, we observe similar behaviors to support our comments. Overall, we report an average F1 score of 80% with five points of standard deviation with cross-validation on our better performing models. Note that these F1 values are approximately 10 points less than our reported test results. This is due to test sizes getting one-fifth of the original size with fivefold validation and even less with 10-fold case. Considering that *O* entities are much more frequent than the others (Table 3), having more *O* tag ratio on a partition by chance dramatically reflects on the performance since sample sizes of the other entity types shrink substantially to compensate. Still, the cross-validation results support our claims about the relationship between the proposed models.

Statistical significance analyses are carried out for our experimental results. It is once again important to stress that our aim is to present a baseline for Turkish Legal NER domain rather than introducing state-of-the-art architectures. Therefore, we present these metrics to lay a better baseline. Replicating the experiments 10 times per model, we conducted analysis of variance (ANOVA) tests on the results. Overall, our total of nine models represent the variance in data quite well with

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 10. Statistical significance of the experimental results presented in Table 8. "+" for p-value < 0.05, "–" otherwise

	LC-GloVe	LC-Morph2vec	LC-Hybrid	LLC-GloVe	LLC-Morph2vec	LLC-Hybrid	CLC-GloVe	CLC-Morph2vec
LC-Morph2vec	+							
LC-Hybrid	–	+						
LLC-GloVe	+	+	+					
LLC-Morph2vec	+	+	+	+				
LLC-Hybrid	+	+	+	–	+			
CLC-GloVe	+	+	+	–	+	–		
CLC-Morph2vec	+	+	+	+	+	+	+	
CLC-Hybrid	+	+	+	–	–	–	–	+

$p < 0.05$ value. Combining them in groups of three as LSTM-CRF, LSTM-LSTM-CRF, and CNN-LSTM-CRF-based architectures, we once again obtain statistically significant ANOVA results for respective embeddings within these groups. To compare the models one by one with respect to nine-factor experiments, confidence intervals of LC-based models are negative and contain zeros. It follows that we can conclude LSTM-CRF models do not contribute much to the goodness-of-fit to the data at hand. We further conducted t-tests to the performances (as given in Table 8) of dual combinations of our models. Whether the dual combinations are statistically significant or not are presented in Table 10. Out of 36 possible combinations, we find 28 statistically significant results, whereas only 8 insignificant changes on the outcomes are detected, most of which occurring on cross-architectural comparisons. One common pattern is between GloVe and Hybrid embeddings for each architecture, where we find $p > 0.05$ for all of them. Therefore, we will qualitatively analyze some features for which Hybrid representations capture better than GloVe ones in the next subsection.

5.2 Qualitative results

In order to provide qualitative analysis on our results, we present sample test sentences. These sentences are representative examples of the enhancing effect of Hybrid representations over GloVe only ones. The first sentence is tested on LSTM-LSTM-CRF and the second one is tested on CNN-LSTM-CRF models. These sentences are provided in Table 11. Their correct tagging and corresponding predictions of GloVe and Hybrid embedded models are given in Tables 12 and 13, respectively. We selected these empirically better models, which have poor significance test results, for qualitative comparison to keep this section succinct.

In Table 12, LSTM-LSTM-CRF-Hybrid is able to classify both forms of *Hazine* (Treasure), but LSTM-LSTM-CRF GloVe can only classify the orientation form of the word. This is a result of the Morph2Vec portion of the Hybrid representation that learns the suffix structures.

5.3 Comparison with related works on NER tasks in general domain

In this subsection, we evaluate our architectures on two different corpora and compare our results with previously published results. First of them is the CONLL2003 corpus in English, which contains around 300K tokens. The other is the Turkish corpus presented in Tür *et al.* (2003), which consists around 350K tokens. Detailed properties of both corpora are given in Table 14. Even though succeeding works alter them in minor ways, both corpora are most widely used ones in their respective languages.

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 11. Example test sentences

Sentence 1:	TMK.nn 713/2. Maddesindeki bilinmeme nedenine dayalı davalarda Hazinenin davada taraf durumunu alması ve davanın Hazineye yöneltilmesi gerekir.
Translation:	*In actions filed for "reason of being unknown" based on Article 713/2 of the Turkish Civil Code, the Treasury should become a party in proceedings and the action should be directed towards the Treasury.*
Sentence 2:	Mahkemece tekerrüre esas alınan dosyada mevcut bulunan Ödemiş Sulh Ceza Mahkemesinin 2011/1055 – 2012/245 sayılı TCK'nın 125/1-3-a maddeleri gereğince . . .
Translation:	*(Decision of) . . .Criminal Court of Peace of Ödemiş province of 2011/1055 - 2012/245 based on Articles 125/1-3-a of the Turkish Criminal Code which is found in the case file that is taken as a basis for repetition by the Court. . .*

Table 12. Sentence 1 prediction comparison

Words	LLC GloVe	LLC Hybrid	Ground Truth
TMK.nın	B-LEG	B-LEG	B-LEG
713/2	I-LEG	I-LEG	I-LEG
.	I-LEG	I-LEG	I-LEG
Maddesindeki	I-LEG	I-LEG	I-LEG
bilinmeme	O	O	O
nedenine	O	O	O
dayalı	O	O	O
davalarda	O	O	O
Hazinenin	**O**	**B-ORG**	**B-ORG**
davada	O	O	O
taraf	O	O	O
durumunu	O	O	O
alması	O	O	O
ve	O	O	O
davanın	O	O	O
Hazineye	**B-ORG**	**B-ORG**	**B-ORG**
yöneltilmesi	O	O	O
gerekir	O	O	O
.	O	O	O

The results of our models and the performances of several baseline methods on CONLL2003 and Tür *et al.* (2003) corpora are presented in Table 15. These results should be investigated in two dimensions, English and Turkish. In the case of English, we obtain highly comparable results. Despite the fact that English is not a morphologically rich language, even the architectures including pure Morph2Vec embeddings have acceptable results. Results on Turkish NER, on the other

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 13. Sentence 2 prediction comparison

Words	CLC GloVe	CLC Hybrid	Ground Truth
Mahkemece	O	O	O
tekerrüre	O	O	O
esas	O	O	O
alınan	O	O	O
dosyada	O	O	O
mevcut	O	O	O
bulunan	O	O	O
Ödemiş	B-COU	B-COU	B-COU
Sulh	I-COU	I-COU	I-COU
Ceza	I-COU	I-COU	I-COU
Mahkemesinin	I-COU	I-COU	I-COU
201/1055	B-REF	B-REF	B-REF
–	I-REF	I-REF	I-REF
2012/245	**B-LEG**	**I-REF**	**I-REF**
sayılı	**I-LEG**	**I-REF**	**I-REF**
TCK'nın	**I-LEG**	**B-LEG**	**B-LEG**
125/1-3-a	I-LEG	I-LEG	I-LEG
maddeleri	I-LEG	I-LEG	I-LEG
gereğince	O	O	O

Table 14. Properties of corpora used in comparisons

	CONLL2003			Tür *et al.* (2003) corpus		
	Training	*Test*	*Total*	*Training*	*Test*	*Total*
PER	8361	1597	9958	9317	1176	10,493
LOC	7596	1664	9260	6777	913	7690
ORG	7596	1654	9250	6267	673	6940
MISC	8957	698	9655	–	–	–
O	212,337	38,323	250,660	283,147	33,226	316,373
Total	254,985	46,436	301,421	314,866	37,301	352,167

hand, are slightly trickier to investigate. The Turkish corpus had several modifications since its first publication in 2003. We had access to the version of Güngör *et al.* (2019) which includes morphological descriptions of each word in the corpus, where their notation was introduced by Oflazer (1994). Their morphological analyzer pipeline with handcrafted linguistic rules utilize this

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Table 15. Comparison of the performance of our approach with other NER corpora

Models for English NER	CONLL2003	Models for Turkish NER	Tür *et al*. (2003) corpus
Lample *et al*. (2016)	90.94	Şeker and Eryiğit (2012)	91.94
Ma and Hovy (2016)	91.21	Demir and Özgür (2014)	91.85
Chiu and Nichols (2016)	91.62	Kuru *et al*. (2016)	91.30
Peters *et al*. (2018)	92.22	Akdemir (2018)	90.90
Straková *et al*. (2019)	93.38	Güngör *et al*. (2019)	**92.93**
Yamada *et al*. (2020)	94.30	–	–
Wang *et al*. (2020)	**94.60**	–	–
LSTM-CRF (GloVe)	92.83	LSTM-CRF (GloVe)	61.65
LSTM-CRF (Morph2Vec)	87.05	LSTM-CRF (Morph2Vec)	61.02
LSTM-CRF (Hybrid)	91.27	LSTM-CRF (Hybrid)	64.59
LSTM-LSTM-CRF (GloVe)	93.28	LSTM-LSTM-CRF (GloVe)	86.12
LSTM-LSTM-CRF (Morph2Vec)	89.67	LSTM-LSTM-CRF (Morph2Vec)	84.40
LSTM-LSTM-CRF (Hybrid)	92.63	LSTM-LSTM-CRF (Hybrid)	**89.06**
CNN-LSTM-CRF (GloVe)	**93.57**	CNN-LSTM-CRF (GloVe)	85.22
CNN-LSTM-CRF (Morph2Vec)	91.37	CNN-LSTM-CRF (Morph2Vec)	83.67
CNN-LSTM-CRF (Hybrid)	92.78	CNN-LSTM-CRF (Hybrid)	87.48

additional information and substantially increase the performance of the architecture. We had to exclude these part of the corpus to make it compatible with our architectures. Morph2Vec, which was trained in an unsupervised manner, is not able to match this morphological analyzer. That being said, the scope is not to improve the state-of-the-results neither in the Turkish nor in English "general" NER but to set the baseline in Turkish legal domain NER. Nevertheless, our results on these domains are also quite comparable with the state-of-the-art results reported in the literature. As a final note, one of the possible future research directions would be to enhance our legal NER corpus with the morphological descriptions as Güngör *et al*. (2019) did for the general Turkish NER and then combine Güngör *et al*. (2019)'s approach with ours to obtain better performing methods.

6. Discussion

Among the proposed architectures, the best performance is obtained from LLC model fed with Hybrid (GloVe + Morph2Vec) word representation, reaching 92.27% overall F1 score. As it is mentioned in Section 3, this model is LC-based, with an additional second LSTM layer added for configuring character representations. The initial expectation that character representations perform better than others for an agglutinative language cannot not be extended to Morph2Vec results. This is caused by the construction methodology of Morph2Vec embeddings. There are two different techniques to create Morph2Vec, one with supervised segmentation and the other with unsupervised segmentation of a word. In Üstün *et al*. (2018), it is reported that the supervised technique produces better results as expected. However, due to the lack of previously segmented legal words and considering the cost of obtaining one, we had to deploy unsupervised method.

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Composing a legal corpus with segmented legal words can be an important future work to improve performance.

We now turn to the detailed comparisons between word embeddings and fix the architecture to the best-performing LLC model. When we compare performances of Morph2Vec embeddings in LLC model compared to the baseline GloVe (F1 scores betwen Tables 5 and 6), improvements were observed only for LOC and COU entities. Additionally, the decline of overall F1 score from 92.18% to 90.06% when Morph2Vec is used single-handedly to feed LLC model is irrefutable. When we compare the performances of hybrid embeddings (GloVe + Morph2Vec) with those of GloVe in LLC architecture (F1 scores betwen Tables 5 and 7), we observe improvements for entities DAT (93.19% to 93.23%) and LEG (89.90% to 90.91%) in a similar fashion. In general, a remarkable increase in performance is obtained with final F1 score of 92.27%. This means, besides the readily improved entities, Hybrid embeddings also are on par with GloVe for the remaining entities, that is, 86.18% in REF (86.18% in GloVe) and 96.99% in PER (97.00% in GloVe).

A similar behavior can be observed for the remaining NER models. For CLC, replacing GloVe to Morph2Vec as the base word embedding only improved LEG class slightly with 0.42%. On the other hand, the performance decreases for other entities are considerably large with scores ranging up to 25.00% (LOC). The same substitution of embeddings do not improve any of the entities in the LC model as well, where a drastic deterioration for ORG class (57.45% to 40.00%) is observed. The steepest decrease in total F1 score happens in LC model with 2.82%.

When it comes to the inter-model comparisons, according to emprical results, LSTM-LSTM-CRF-GloVe and LSTM-LSTM-CRF-Hybrid representation perform better in overall, while CNN-LSTM-CRF performs better with Morph2Vec. However, according to the further statistical tests we have conducted, it turns out that results from the experiments with LSTM-LSTM-CRF-GloVe, LSTM-LSTM-CRF-Hybrid, CNN-LSTM-CRF-GloVe and CNN-LSTM-CRF-Hybrid are not statistically significant, since the results of pair-wise t-test is higher than 0.05. This means that these architectures can be treated equally well to be applied in the legal domain. On the other hand, if performances across entities are investigated, it is observed that CNN-LSTM-CRF model results in half of the best performances across different base word representations. Within the results for CNN-LSTM-CRF model, GloVe reaches the best ones for ORG (83.33%) and REF (87.80%), where Morph2Vec also reaches 87.80% for the REF class. CNN-LSTM-CRF model gives the best results in PER (97.28%) and COU (92.65%) with Hybrid base representations. The remaining half of the best results are achieved by LSTM-LSTM-CRF model. We can argue that the performance of LSTM-LSTM-CRF models and CNN-LSTM-CRF models fluctuate among classes. However, these two models are clearly superior to LSTM-CRF, which is expected due to the lack of aforementioned consideration of sub-word information.

Upon inspecting results, one notices the considerably low performances for LOC and considerably high performances for OFF entities. Referring to Table 3, both of LOC and OFF have low total counts with respect to the other entities with 51 and 88 occurrences in the corpus, respectively. Although *location*s are expected to be frequently mentioned in legal texts and case documents, they are usually referred in court names belonging to names of provinces and cities such as *Ankara 1. Asliye Mahkemesi* (*First Chamber of Ankara Civil Court of First Instance*) refers to a specific court in the province "Ankara." There are only five occurrences of LOC class in our test set. As a result, low number of training samples causes the deteriorated performance for LOC entity. On the other extreme, the typical form of *official gazette*, where their publication dates and issue numbers are presented, make this legal entity easily "learnable" by our models.

7. Conclusion

In this paper, we have studied NER models in the legal domain for Turkish language. We have proposed several neural network-based architectures along with several word embedding models to address NER in Turkish legal texts. Our models utilize different combinations of various

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

tools, such as BiLSTM, CRF, and CNN models with character-level features as well as Morph2Vec and GloVe embeddings and their combinations. We contributed by compiling and providing a Turkish NER corpus in the legal domain and a NE set in the legal domain, which is manually labeled on our corpus. In addition to the common NEs, *person, location, organization*, and *date*, we have created our custom NEs in the legal domain as *legislation, reference, court name*, and *official gazette*. We have comprehensively studied the Turkish NER in the legal domain and have set baseline results. We reached the best performance when LLC model is fed by our proposed hybrid word representation containing Morph2Vec and GloVe embeddings and supported by character-level features extracted with BiLSTM. By using this combination of model and word representations, we reported upto 92.27% F1 score in overall performance.

As future work, better results can be obtained by extending the corpus either in number or by including documents from various courts. An extension to our corpus can also pave the way to spare some data as the development set, which is very useful to tune hyperparameters and obtain better results. Another way to improve these results is to include the morphological structures of words. This way, Morph2Vec can be trained in a supervised fashion, which can have a substantial effect on the overall performance.

Acknowledgment. This work was supported by TUBITAK 1001 grant (120E346). We thank Barş Özçelik of the School of Law, Bilkent University, for precious consultation on law. We would like to also thank the anonymous reviewers for their many insightful comments which significantly improved our manuscript.

References

Akdemir A. (2018). *Named Entity Recognition in Turkish Using Deep Learning Methods and Joint Learning*. PhD Thesis, Bogaziçi University.

Akkaya E.K. and Can B. (2021). Transfer learning for Turkish named entity recognition on noisy text. *Natural Language Engineering* **27**(1), 35–64.

Aletras N., Tsarapatsanis D., PreoŢiuc-Pietro D. and Lampos V. (2016). Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ Computer Science* **2**, e93.

Aleven V. (2003). Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment. *Artificial Intelligence* **150**, 183–237.

Ashley K.D. and Brüninghaus S. (2009). Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law* **17**(2), 125–165.

Bach N.X., Minh N.L., Oanh T.T. and Shimazu A. (2013). A two-phase framework for learning logical structures of paragraphs in legal articles. *ACM Transactions on Asian Language Information Processing* **12**(1), 1–32.

Bench-Capon T., Araszkiewicz A.M., Ashley A.K., Atkinson K., Bex F., Borges F., Bourcier D., Bourgine P., Conrad J.G., Francesconi E., Gordon T.F., Governatori G., Leidner J.L., Lewis D.D., Loui R.P., McCarty L.T., Prakken H., Schilder F., Schweighofer E., Thompson P., Tyrrell A., Verheij, B., Walton D.N. and Wyner A.Z. (2012). A history of AI and Law in 50 papers: 25 years of the international conference on AI and Law. *Artificial Intelligence and Law* **20**, 215–319.

Borthwick A. and Grishman R. (1999). *A Maximum Entropy Approach to Named Entity Recognition*. PhD Thesis, Citeseer.

Branting K.L., Yeh A., Weiss B., Merkhofer E. and Brown B. (2018). Inducing predictive models for decision support in administrative adjudication. In Pagallo U., Palmirani,M., Casanovas P., Sartor G. and Villata, S. (eds), *AI Approaches to the Complexity of Legal Systems*. Springer International Publishing, pp. 465–477.

Buchanan B.G. and Headrick T.E. (1970). Some speculation about artificial intelligence and legal reasoning. *Stanford Law Review* **23**, 40–62.

Cardellino C., Teruel M., Alemany L.A. and Villata S. (2017). A low-cost, high-coverage legal named entity recognizer, classifier and linker. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, pp. 9–18.

Casanovas P., Pagallo U., Palmirani M. and Sartor G. (eds) (2014). *AI Approaches to the Complexity of Legal Systems - AICOL 2013 International Workshops, AICOL-IV@IVR, Belo Horizonte, Brazil, July 21–27, 2013 and AICOL-V@SINTELNET-JURIX, Bologna, Italy, December 11, 2013, Revised Selected Papers*. Lecture Notes in Computer Science. Springer International Publishing.

Chalkidis I., Androutsopoulos I. and Michos A. (2017). Extracting contract elements. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, pp. 19–28.

Chalkidis I., Fergadiotis E., Malakasiotis P., Aletras N. and Androutsopoulos I. (2019a). Extreme multi-label legal text classification: A case study in EU legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, Minneapolis, Minnesota. Association for Computational Linguistics, pp. 78–87.

Chalkidis I., Fergadiotis E., Malakasiotis P. and Androutsopoulos I. (2019b). Large-scale multi-label text classification on EU legislation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics, pp. 6314–6322.

Chalkidis I., Jana A., Hartung D., Bommarito M.J., Androutsopoulos I., Katz D.M. and Aletras N. (2021). LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. Available at SSRN 3936759.

Chalkidis I. and Kampas D. (2019). Deep learning in law: Early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law* **27**(2), 171–198.

Chiu J.P. and Nichols E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* **4**, 357–370.

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K. and Kuksa P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research* **12**(ARTICLE), 2493–2537.

Dale R. (2019). Law and word order: NLP in legal tech. *Natural Language Engineering* **25**, 211–217.

Dalkılıç F.E., Gelişli S. and Diri B. (2010). Named entity recognition from Turkish texts. In *2010 IEEE 18th Signal Processing and Communications Applications Conference*. IEEE, pp. 918–920.

Demir H. and Özgür A. (2014). Improving named entity recognition for morphologically rich languages using word embeddings. In *2014 13th International Conference on Machine Learning and Applications*. IEEE, pp. 117–122.

Dozier C., Kondadadi R., Light M., Vachher A., Veeramachaneni S. and Wudali R. (2010). Named entity recognition and resolution in legal text. In *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*. Berlin, Heidelberg: Springer-Verlag, pp. 27–43.

Elnaggar A., Otto R. and Matthes F. (2018). Deep learning for named-entity linking with transfer learning for legal documents. In *Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference*, pp. 23–28.

Eryiğit G. (2014). Itu turkish nlp web service. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1–4.

Evans R., Piwek P., Cahill L. and Tipper N. (2008). Natural language processing in CLIME, a multilingual legal advisory system. *Natural Language Engineering* **14**(1), 101–132.

Finkel J.R., Grenager T. and Manning C.D. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 363–370.

Finkel J.R. and Manning C.D. (2009). Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 326–334.

Francesconi E., Montemagni S., Peters W. and Tiscornia D. (eds) (2010). *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*. *Lecture Notes in Computer Science*, vol. 6036. New York, NY: Springer.

Freitag D. and McCallum A. (1999). Information extraction with HMMs and shrinkage. In *Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*. Orlando, Florida, pp. 31–36.

Galgani F., Compton P. and Hoffmann A. (2012). Combining different summarization techniques for legal text. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, HYBRID'12, USA. Association for Computational Linguistics, pp. 115–123.

Graves A. and Schmidhuber J. (2005). Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005*, vol. 4. IEEE, pp. 2047–2052.

Grishman R. and Sundheim B. (1995). Design of the MUC-6 evaluation. In *Proceedings of the 6th Conference on Message Understanding*. Association for Computational Linguistics, pp. 1–11.

Güngör O., Güngör T. and Üsküdarli S. (2019). The effect of morphology in named entity recognition with sequence tagging. *Natural Language Engineering* **25**(1), 147–169.

Hakkani-Tür D.Z., Oflazer K. and Tür G. (2002). Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities* **36**(4), 381–410.

Hammerton J. (2003). Named entity recognition with long short-term memory. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL'03, USA. Association for Computational Linguistics, pp. 172–175.

Hochreiter S. and Schmidhuber J. (1997). Long short-term memory. *Neural Computation* **9**(8), 1735–1780.

Huang Z., Xu W. and Yu K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

Ikram A.Y. and Chakir L. (2019). Arabic text classification in the legal domain. In *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*. IEEE, pp. 1–6.

Jackson P., Al-Kofahi K., Tyrrell A. and Vachher A. (2003). Information extraction from case law and retrieval of prior cases. *Artificial Intelligence* **150**, 239–290.

Kanapala A., Pal S. and Pamula R. (2019). Text summarization from legal documents: A survey. *Artificial Intelligence Review* **51**, 371–402.

Katz D.M., Bommarito M.J. and Blackman J. (2017). A general approach for predicting the behavior of the supreme court of the United States. *PloS One* **12**(4), e0174698.

Kim M.-Y., **Xu Y. and Goebel R.** (2017). Applying a convolutional neural network to legal question answering. In Otake M., Kurahashi S., Ota Y., Satoh K. and Bekki D. (eds), *New Frontiers in Artificial Intelligence*. Springer International Publishing, pp. 282–294.

Kowsrihawat K., **Vateekul P. and Boonkwan P.** (2018). Predicting judicial decisions of criminal cases from Thai Supreme Court using bi-directional GRU with attention mechanism. In *2018 5th Asian Conference on Defense Technology (ACDT)*, pp. 50–55.

Krishnan V. and Manning C.D. (2006). An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, USA. Association for Computational Linguistics, pp. 1121–1128.

Küçük D. and Steinberger R. (2014). Experiments to improve named entity recognition on Turkish tweets. arXiv preprint arXiv:1410.8668.

Küçük D. and Yazıcı A. (2009). Named entity recognition experiments on Turkish texts. In Andreasen T., Yager R.R., Bulskov H., Christiansen H. and Larsen H.L. (eds), *Flexible Query Answering Systems*, Berlin, Heidelberg. Springer, pp. 524–535.

Küçük D. and Yazıcı A. (2012). A hybrid named entity recognizer for Turkish. *Expert Systems with Applications* **39**(3), 2733–2742.

Kuru O., **Can O.A. and Yuret D.** (2016). CharNER: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 911–921.

Lafferty J.D., **McCallum A. and Pereira F.C.N.** (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML'01, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., pp. 282–289.

Lample G., **Ballesteros M.**, **Subramanian S.**, **Kawakami K. and Dyer C.** (2016). Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360.

Leek T.R. (1997). *Information Extraction using Hidden Markov Models*. Master's Thesis, Citeseer.

Leitner E., **Rehm G. and Moreno-Schneider J.** (2019). Fine-grained named entity recognition in legal documents. In *International Conference on Semantic Systems*. Springer, pp. 272–287.

Li J., **Sun A.**, **Han J. and Li C.** (2022). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering* **34**(1), 50–70.

Lim C., **Tan I. and Selvaretnam B.** (2019). Domain-general versus domain-specific named entity recognition: A case study using TEXT. In *Multi-disciplinary Trends in Artificial Intelligence, 13th International Conference, MIWAI 2019, Kuala Lumpur, Malaysia*, pp. 238–246.

Long S., **Tu C.**, **Liu Z. and Sun M.** (2019). Automatic judgment prediction via legal reading comprehension. In *China National Conference on Chinese Computational Linguistics*. Springer, pp. 558–572.

Luz de Araujo P.H., **de Campos T.E.**, **de Oliveira R.R.R.**, **Stauffer M.**, **Couto S. and Bermejo P.** (2018). LeNER-Br: A dataset for named entity recognition in Brazilian legal text. In *International Conference on the Computational Processing of Portuguese (PROPOR)*, Lecture Notes on Computer Science (LNCS), Canela, RS, Brazil. Springer, pp. 313–323.

Ma X. and Hovy E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint arXiv:1603.01354.

Manor L. and Junyi J.L. (2019). Plain English summarization of contracts. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pp. 1–11.

Martin A.D., **Quinn K.M.**, **Ruger T.W. and Kim P.T.** (2004). Competing approaches to predicting supreme court decision making. *Perspectives on Politics* **2**(4), 761–767.

McCallum A., **Freitag D. and Pereira F.C.N.** (2000). Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*, ICML'00, San Francisco, CA, USA, pp. 591–598.

Medvedeva M., **Vols M. and Wieling M.** (2020). Using machine learning to predict decisions of the European Court of Human Rights. *Artificial Intelligence and Law* **28**(2), 237–266.

Mikolov T., **Chen K.**, **Corrado G. and Dean J.** (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Morimoto A., **Kubo D.**, **Sato M.**, **Shindo H. and Matsumoto Y.** (2017). Legal question answering system using neural attention. In Satoh K., Kim M., Kano Y., Goebel R. and Oliveira T. (eds), *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), Held in Conjunction with the 16th International Conference on Artificial Intelligence and Law (ICAIL 2017) in King's College London, UK*. EPiC Series in Computing, vol. 47. EasyChair, pp. 79–89.

Mumcuoğlu E., **Öztürk C.E.**, **Ozaktas H.M. and Koç A.** (2021). Natural language processing in law: Prediction of outcomes in the higher courts of Turkey. *Information Processing & Management* **58**(5), 102684.

Nanda R., **Adebayo K.J.**, **Di Caro L.**, **Boella G. and Robaldo L.** (2017). Legal information retrieval using topic clustering and neural networks. In *COLIEE@ ICAIL*, pp. 68–78.

Nguyen T.-S., **Nguyen L.-M.**, **Tojo S.**, **Satoh K. and Shimazu A.** (2018). Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts. *Artificial Intelligence and Law* **26**(2), 169–199.

Oflazer K. (1994). Two-level description of Turkish morphology. *Literary and Linguistic Computing* **9**(2), 137–148.

https://doi.org/10.1017/S1351324922000304 Published online by Cambridge University Press

Oflazer K. and Saraçlar M. (eds) (2018). *Turkish Natural Language Processing*, New York: Springer.

O'Sullivan C. and Beel J. (2019). Predicting the outcome of judicial decisions made by the European Court of Human Rights. In *Proceedings of the 27th AIAI Irish Conference on Artificial Intelligence and Cognitive Science*, Dublin, Ireland.

Pennington J., Socher R. and Manning C.D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Peters M.E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K. and Zettlemoyer L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Ruger T.W., Kim P.T., Martin A.D. and Quinn K.M. (2004). The supreme court forecasting project: Legal and political science approaches to predicting supreme court decisionmaking. *Columbia Law Review*, **104**(4), 1150–1210.

Sang E.F. and De Meulder F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050.

Sangeetha D., Kavyashri R., Swetha S. and Vignesh S. (2017). Information retrieval system for laws. In *2016 Eighth International Conference on Advanced Computing (ICoAC)*. IEEE, pp. 212–217.

Şeker G.A. and Eryiğit G. (2012). Initial explorations on using CRFs for Turkish named entity recognition. In *Proceedings of COLING 2012*, pp. 2459–2474.

Sevim N., Şahinuç F. and Koç A. (2022). Gender bias in legal corpora and debiasing it. *Natural Language Engineering*, 1–34.

Shulayeva O., Siddharthan A. and Wyner A. (2017). Recognizing cited facts and principles in legal judgements. *Artificial Intelligence and Law* **25**(1), 107–126.

Simonson D., Broderick D. and Herr J. (2019). The extent of repetition in contract language. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pp. 21–30.

Sleimi A., Sannier N., Sabetzadeh M., Briand L. and Dann J. (2018). Automated extraction of semantic legal metadata using natural language processing. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*. IEEE, pp. 124–135.

Soh J., Lin H.K. and Chai I.E. (2019). Legal area classification: A comparative study of text classifiers on Singapore Supreme Court Judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pp. 67–71.

Straková J., Straka M. and Hajič J. (2019). Neural architectures for nested NER through linearization. arXiv preprint arXiv:1908.06926.

Şulea O., Zampieri M., Malmasi S., Vela M., Dinu L.P. and van Genabith J. (2017a). Exploring the use of text classification in the legal domain. In Ashley K.D., Atkinson K., Branting L.K., Francesconi, E., Grabmair, M., Lauritsen M., Walker V.R. and Wyner A.Z. (eds), *Proceedings of the Second Workshop on Automated Semantic Analysis of Information in Legal Texts Co-located with the 16th International Conference on Artificial Intelligence and Law (ICAIL 2017), London, UK, June 16, 2017*. CEUR Workshop Proceedings, vol. 2143. CEUR-WS.org.

Şulea O.-M., Zampieri M., Vela M. and van Genabith J. (2017b). Predicting the law area and decisions of French Supreme Court cases. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, Varna, Bulgaria, pp. 716–722.

Tkachenko M. and Simanovsky A. (2012). Named entity recognition: Exploring features. In *KONVENS*, pp. 118–127.

Tür G., Hakkani-Tür D. and Oflazer K. (2003). A statistical information extraction system for turkish. *Natural Language Engineering* **9**(2), 181–210.

Üstün A., Kurfal M. and Can B. (2018). Characters or morphemes: How to represent words? In *Proceedings of The Third Workshop on Representation Learning for NLP*, pp. 144–153.

Vardhan H., Surana N. and Tripathy B. (2020). Named-entity recognition for legal documents. In *International Conference on Advanced Machine Learning Technologies and Applications*. Springer, pp. 469–479.

Virtucio M.B., Aborot J., Abonita J.K., Aviñante R., Copino R.J., Neverida M., Osiana V., Peramo E., Syjuco J. and Tan G.B. (2018). Predicting decisions of the Philippine supreme court using natural language processing and machine learning. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, pp. 130–135.

Wang X., Jiang Y., Bach N., Wang T., Huang Z., Huang F. and Tu K. (2020). Automated Concatenation of Embeddings for Structured Prediction. arXiv preprint arXiv:2010.05006.

Yadav V. and Bethard S. (2019). A survey on recent advances in named entity recognition from deep learning models. arXiv preprint arXiv:1910.11470.

Yamada I., Asai A., Shindo H., Takeda H. and Matsumoto Y. (2020). Luke: Deep contextualized entity representations with entity-aware self-attention. arXiv preprint arXiv:2010.01057.

Yeniterzi R. (2011). Exploiting morphology in Turkish named entity recognition system. In *Proceedings of the ACL 2011 Student Session*, pp. 105–110.

Yeniterzi R., Tür G. and Oflazer K. (2018). Turkish named-entity recognition. In *Turkish Natural Language Processing*. Springer, pp. 115–132.

Cite this article: Çetindağ C, Yazıcıoğlu B and Koç A (2023). Named-entity recognition in Turkish legal texts. *Natural Language Engineering* **29**, 615–642. https://doi.org/10.1017/S1351324922000304