NER sur décisions judiciaires françaises : CamemBERT Judiciaire ou méthode ensembliste ?

Sid Ali Mahmoudi*, Charles Condevaux*, Bruno Mathis**, Guillaume Zambrano*, Stéphane Mussard*

*Univ. Nîmes CHROME, Avenue du Dr Georges Salan, 30000 Nîmes
sid.mahmoudi@unimes.fr,
https://chrome.unimes.fr
**Centre Européen de Droit et d'Economie, ESSEC Business School,
3 Av. Bernard Hirsch, 95000 Cergy
bmathis@cegetel.net

Résumé. Nous étudions dans cet article les apports respectifs de différentes représentations de mots, de la méthode ensembliste et d'un *transformer* spécialisé que nous appelons CamemBERT judiciaire, sur la tâche de recherche d'entités nommées dans les décisions de justice françaises. Nous comparons les performances des modèles BiLSTM-CRFs entre eux, individuellement ou constitués en ensembles, et avec le modèle de Ngompé et al. (2019) pris comme référence à battre. Les résultats obtenus montrent une amélioration.

1 Introduction

La première expérimentation d'apprentissage automatique de décisions de justice a été conduite par Aletras et al. (2016). Elle a montré dans quelle mesure la machine pouvait parvenir à la même décision que les juges de la Cour européenne des droits de l'homme avec un algorithme SVM. Suárez et al. (2020) ont montré que depuis lors, les approches neuronales ont considérablement amélioré l'état de l'art dans le traitement automatique du langage naturel en général et dans la reconnaissance d'entités nommées (NER) en particulier.

La Cour de cassation française a été la première à appliquer la NER à des décisions de justice françaises. Barriere et Fouret (2019) ont, pour elle, annoté des arrêts avec 4 étiquettes, personne physique, personne morale, adresse et date de naissance, et ont fait entraîner ces annotations à un réseau de neurones. La reconnaissance d'entités fonctionne sur tout type de chaîne de caractères. Filtz et al. (2020) l'ont appliquée à des dates, et Fernandes et al. (2020) à des montants, dans des décisions judiciaires rédigées respectivement en allemand et en portugais. Mathis (2021) a étendu cette approche à de nombreuses autres entités liées à la procédure judiciaire, soit un total de 27 étiquettes, comme la juridiction et les dates de jugement. Sur un large ensemble de données (1706 décisions), il a évalué les champs aléatoires conditionnels (CRF), de Lafferty et al. (2001), SpaCy, de Honnibal et Montani (2017), Flair, d'Akbik et al. (2019), et DeLFT (Deep Learning For Text), de Lopez (2018).

La sortie en 2018 par Google de son modèle BERT, de Devlin et al. (2018), basé sur un vaste corpus généraliste, a été suivie de variantes sectorielles. Le modèle dit Legal BERT, de

Chalkidis et al. (2020), a été expérimenté pour de la classification, mais pas pour de la NER. Il montre qu'un pré-entraînement mené à partir de rien sur du contenu juridique uniquement est plus efficace qu'un pré-entraînement complémentaire après ajout de contenu juridique à la base BERT d'origine. La plupart des recherches ont porté sur de l'anglais. CamemBERT, de Martin et al. (2019), et FlauBERT, de Le et al. (2019), sont les plus connus des modèles linguistiques généralistes français dérivés de BERT. JuriBERT, de Douka et al. (2021), est une variante de CamemBERT spécialisée dans le droit. Comme Legal BERT, il s'appuie sur un corpus et un nombre d'étiquettes réduits. Le but de cet article est d'évaluer quel gain en performance par rapport au modèle CRF (*baseline*) de Ngompé et al. (2019) peut être obtenu, d'une part avec un CamemBERT Judiciaire, d'autre part en recourant à des méthodes d'ensemble. Les objectifs de cet article sont les suivants.

1. Nous proposons un CamemBERT Judiciaire afin de vérifier si ses performances vont au-delà du modèle CRF (*baseline*) de Ngompé et al. (2019).

2. Nous montrons que les méthodes d'ensemble BiLSTM-CRF peuvent aussi donner de meilleurs résultats que la *baseline*, en utilisant à la fois des *embeddings* (plongements) lexicaux statiques et contextuels.

2 État de l'art

2.1 BiLSTM-CRF, transformers et NER

Li et al. (2020) définissent la reconnaissance d'entités nommées (NER) comme la reconnaissance des mots ou séquences de mots appartenant à des types sémantiques prédéfinis tels que la personne, le lieu, l'organisation, etc. ou liées au domaine d'intérêt, comme, ici, le type de juridiction. Les approches traditionnelles de NER comprennent l'exécution de règles, l'apprentissage non supervisé comme le *clustering* et l'apprentissage supervisé comme le HMM et le CRF. Des approches plus récentes utilisent les algorithmes d'apprentissage profond comme le LSTM (Hochreiter et Schmidhuber, 1997) et sa variante bidirectionnelle avec une couche CRF (BiLSTM-CRF) (Huang et al., 2015)), qui avance et recule le long des phrases pour une meilleure appréhension du contexte. D'autres s'appuient sur des réseaux convolutifs (caractère BiLSTM-CRF + CNN), qui fonctionnent au niveau du caractère plutôt qu'au niveau du mot (*token)*. Enfin, le *transformer* utilise l'auto-attention empilée et des couches entièrement connectées par points pour construire les blocs de base du codeur et du décodeur.

Selon Li et al. (2020), les expériences menées sur diverses tâches montrent que les *transformers*, en particulier BERT, produisent une qualité supérieure tout en prenant beaucoup moins de temps d'apprentissage. Dans le domaine judiciaire, Trias et al. (2021) et Shankar et Buddarapu (2018) ont démontré l'intérêt de la démarche ensembliste appliquée à la NER avec l'apprentissage profond.

2.2 Baseline

Les CRF cherchent à maximiser une probabilité conjointe en se fondant sur l'idée que les étiquettes peuvent être liées à d'autres étiquettes selon leur position dans le document ou une sous-séquence donnée du document.

Ngompé et al. (2019) ont obtenu de très bonnes performances en NER (une *F-measure* globale de 94,82%) avec leur modèle CRF comparé au modèle de type Markov cachés (*HMM*) (McCallum et al., 2000), appliqués aux décisions de justice françaises. La robustesse de leur modèle était basée sur l'ensemble de caractéristiques définies soit manuellement en observant les décisions (lemmatisation des mots, étiquetage grammatical (*POS tagging*), présence de ponctuation ou non, etc.) soit à l'aide des algorithmes BDS et SFFS pour la sélection bidirectionnelle et séquentielle en avant flottante des meilleures caractéristiques. Ces derniers sont implémentés à travers les fonctions potentielles que le modèle essaie de trouver *via* la meilleure pondération possible (les paramètres λ_i) qui maximisent la *F-measure* du modèle durant l'entraînement. Un autre modèle CRF a aussi été entraîné pour découper les décisions en trois sections, pour faciliter la recherche voire améliorer la performance du modèle CRF du NER en réduisant le champ de recherche des entités.

Nous reprenons le jeu de données de Ngompé et al. (2019) et nous comparons leur modèle CRF avec des architectures fondées sur du BiLSTM-CRF avec différents types d'*embeddings*.

3 Méthodologie

3.1 CamemBERT judiciaire

CamemBERT Judiciaire (CJ) est un modèle de type *transformer* dérivé du CamemBERT de Martin et al. (2019). Il a été entraîné sur un ensemble de données de 30 Go constitué de jugements et arrêts d'appel, de la législation, de débats parlementaires et de questions au Gouvernement. Plutôt que d'être entraîné à partir de rien, CJ est entraîné avec un modèle linguistique à base de masque (MLM) à partir du point de contrôle existant de CamemBERT, un modèle généraliste de langue française monté avec la bibliothèque *HuggingFace* de Wolf et al. (2020). Nous faisons porter l'entraînement au niveau du document plutôt que de la phrase, pour augmenter la longueur du contexte. Comme la taille moyenne des documents dépasse de loin la séquence traditionnelle de 512, nous les avons divisés avec un chevauchement de 32 jetons (*tokens*). L'architecture est identique à celle de CamemBERT, à savoir 12 couches, 12 têtes d'attentions et une taille cachée de 768. Le vocabulaire est également inchangé et fixé à 32005 jetons. Pour la phase d'entraînement, nous nous appuyons sur des lots de 2048 phrases, un abandon (*dropout*) de 0,1, sur une phase d'échauffement (*warm-up*) de 500 pas au cours de laquelle le taux d'apprentissage est augmenté linéairement de 0 à 2e-4 puis diminué linéairement au cours de la même époque. Les précisions initiales BPC (*Bits Per Character*) et MLM sont respectivement de 5,079 et 0,518 avant apprentissage et de 0,741 et 0,874 après apprentissage. Cela montre que le langage juridique français est assez prévisible et qu'un simple *transformer* est capable de prédire avec précision les jetons masqués. CJ a été affiné pour la NER avec NERDA, de Kjeldgaard et Nielsen (2021). Il se montre capable de gérer le français standard ainsi que la syntaxe et le vocabulaire spécifiques au domaine juridique.

3.2 Méthode ensembliste BiLSTM-CRF

Le but des modèles d'apprentissage est de trouver la meilleure approximation, l'hypothèse appelée h de la fonction réelle f, qui donne la valeur de y pour chaque observation x telle que $y = f(x)$ pour chaque tuple (x, y) du jeu de données. L'apprentissage ensembliste *(ensemble*

learning) utilise un système de vote pour départager plusieurs modèles dans la prédiction des classes de NER. La figure 1 illustre un exemple de classification à la fois par un modèle typique BiLSTM-CRF et par un ensemble de BiLSTM-CRFs, qu'on appelle apprenants (*learners*) de base. L'entraînement parallèle des apprenants de base fait apparaitre des étiquetages éventuellement différents. Un mécanisme de vote majoritaire permet de les départager. En cas d'égalité entre deux étiquettes, le modèle sélectionne celle de l'apprenant où sa performance est la plus élevée. Nous définissons plusieurs ensembles. Le premier (EO) exploite des *embeddings* sta-

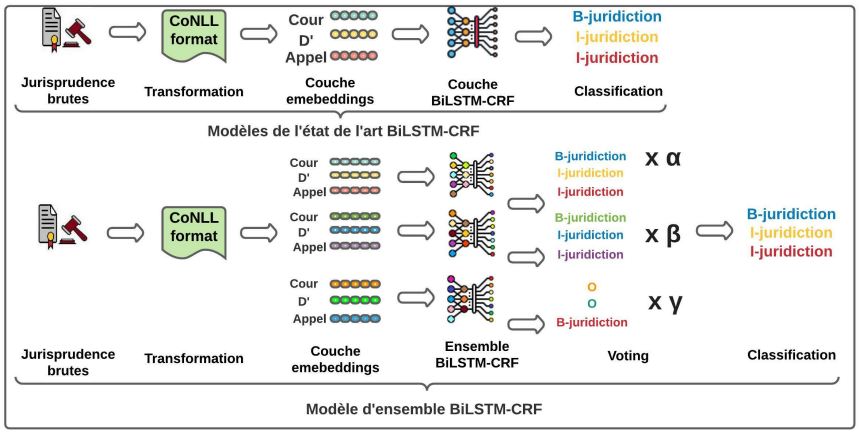

FIG. 1 – *Aperçu de l'ensemble BiLSTM-CRF pour la reconnaissance d'entités nommées en comparaison avec le BiLSTM-CRF standard.*

tiques uniquement (w2vec et FastText), le second (E1) des *embeddings* contextuels (ELMo et BERT-base), le troisième (E2) combine *embeddings* statiques et contextuels. Le dernier (E3) combine tous les modèles BiLSTM-CRF et le CamemBERT judiciaire :

- E0 = BiLSTM-CRF(w2vec) + BiLSTM-CRF(FastText)
- E1 = BiLSTM-CRF(ELMo) + BiLSTM-CRF(BERT-base)
- E2 = E0 + E1
- E3 = E2 + CJ

3.3 Jeu de données utilisé

Afin de tester les performances de la NER sur notre méthode ensembliste et notre CJ, nous utilisons le jeu de données de Ngompé et al. (2019). [1]. Cela permet d'utiliser ses données en entrée et ses résultats comme *baseline*. Ces décisions de justice ont été choisies au hasard en variant les juridictions et les années. Elles comprennent 6 catégories d'entités nommées : le numéro d'enregistrement de la décision (N° de décision), le type de tribunal (Chambre, Juridiction et Ville), la date du jugement (Date) et les personnes telles que les juges, les avocats et

1. Le jeu de données et l'implémentation sont accessibles sur : `https://github.com/lawbot-porject/ensemble_judicial_ner`

les parties (Personne). Les personnes ont été annotées soit par leur nom, soit par leur prénom, soit par leurs initiales en cas de pseudonymisation. La dernière classe est la Fonction de tiers (avocat, juge, conseiller, greffier). Le jeu de données comprend 503 décisions de justice avec un total de 1997384 tokens distribués en 33240 séquences. Dans le jeu de données d'apprentissage, nous observons la distribution suivante des entités nommées (nombre d'annotations) : N° de décision (1321), Chambre (7641), Juridiction (1308), Ville (1304), Date (1590), Personnes (7723), Fonction (2062). [2] Nous avons utilisé le format standard CoNLL-2003 (Sang et De Meulder, 2003) et le modèle BIEO pour représenter les entités (Konkol et Konopík, 2015). Le jeu de données a été réparti entre 80% de données d'entraînement et 20% de données de test.

4 Résultats

4.1 Apprentissages élémentaires

La première tâche de la NER est de représenter le texte. Les *embeddings* statiques comme FastText (Joulin et al., 2016) et Word2vec (Mikolov et al., 2013) permettent de distinguer majuscules et minuscules, noms masculins et féminins, les verbes dans leurs différentes conjugaisons, etc. Les *embeddings* contextuels, comme ELMo (Peters et al., 2018) et le célèbre BERT (Devlin et al., 2018), prennent en compte le contexte des mots dans une phrase lors de la génération de son vecteur de représentation. A partir de DeLFT, nous avons changé la couche d'*embedding* pour chaque apprenant de base. Pour la classification, l'architecture BiLSTM-CRF est utilisée.

Le modèle CRF de Ngompé et al. (2019) constitue la *baseline*. Il obtient d'excellentes F-mesures pour chaque classe, ainsi qu'une bonne F-mesure globale de 94,82%. Le Tableau 1 présente les résultats de chaque modèle élémentaire (en gras les scores supérieurs à ceux de la *baseline*). CJ est très proche de la *baseline* avec une F-mesure globale de 94,43%, mais il ne la surpasse que sur la classe "Personne". Dérivé de CamemBERT, lui-même déjà entraîné sur un large corpus de documents français tels que Wikipédia, CJ a une bonne capacité à prédire les noms propres. De la même manière, chaque apprenant de type BiLSTM-CRF possède une bonne F-mesure globale, et surpasse la *baseline* pour la classe "Personne", sauf ELMo qui sous-performe la *baseline* sur chaque classe.

4.2 Apprentissages ensemblistes

La F-mesure globale augmente avec le nombre d'apprenants, voir Tableau 2. L'ensemble 1, avec des *embeddings* contextuels uniquement, surpasse la *baseline* sur les classes "Personne" et "Fonction", avec une F-mesure globale de 92,50%. La méthode d'Ensemble 0, contenant uniquement des *embeddings* statiques, surpasse la *baseline* sur trois classes.

L'ensemble 2 est construit en ajoutant le modèle BiLSTM-CRF issu de word2vec. Ensuite, la F-mesure sur la classe "Personne" et "Fonction" diminue légèrement mais reste supérieure

2. Les annotations ont été faites par deux juristes. La statistique Kappa de Cohen a été calculée afin de déterminer le taux inter-accord. Un taux Kappa de 0,705 a été obtenu indiquant que le niveau de concordance est considéré comme substantiel puisqu'il se situe dans l'intervalle [0,61-0,80].

	Baseline	CJ	w2vec	FastText	ELMo	BERT-base
Personne	86.72	**89.39**	**94.22**	**89.48**	86.10	**87.01**
Date de décision	97.30	93.03	87.59	88.08	88.81	87.51
Fonction	95.18	94.99	84.17	88.40	86.87	86.21
Citation juridique	99.12	98.05	95.77	88.82	88.49	88.57
Juridiction	99.30	94.67	96.62	88.04	87.78	86.49
N° de décision	97.62	96.77	84.68	88.84	88.55	87.27
Ville	99.04	94.11	81.43	89.26	88.27	87.54
Totalité	94.82	94.43	89.21	88.7	87.84	87.23
Temps d'entraîn.(en s)	230 865	1271	12008	9033	12169	11910

TAB. 1 – *F-mesures des modèles élémentaires*

	Baseline	CJ	E0	E1	E2	E3
Personne	86.72	**89.39**	**93.99**	**93.90**	**90.60**	**98.55**
Date de décision	97.30	93.03	**98.82**	95.48	94.78	95.60
Fonction	95.18	94.99	**97.40**	**91.83**	**96.81**	93.66
Citation juridique	**99.12**	98.05	93.52	93.51	91.75	96.38
Juridiction	**99.30**	94.67	88.42	96.11	92.12	94.21
N° de décision	97.62	96.77	91.77	90.22	**100**	93.33
Ville	99.04	94.11	89.69	86.47	92.12	**100**
Totalité	94.82	94.43	93.37	92.50	93.61	**95.96**

TAB. 2 – *F-mesures des modèles ensemblistes*

à la *baseline*. Néanmoins, la F-mesure est plus élevée sur la classe "N° de décision". L'ensemble 3 obtient le meilleur score : 95.96%, qui s'explique notamment par les très bonnes performances des classes "Personne" (98,55%) et "Ville" (100%).

5 Conclusion

Bien que très bien adapté à la NER, le modèle CRF peut être très lent en pratique. La NER d'ensemble basée sur des modèles BiLSTM-CRF qui mobilisent différents types d'embeddings peut être une bonne alternative, lorsque le nombre d'apprenants de base n'est pas très important. La présence d'un *transformer* comme CJ en apprenant de base, spécialisé dans la tâche désirée, a un impact significatif sur le résultat final.

Références

Akbik, A., T. Bergmann, D. Blythe, K. Rasul, S. Schweter, et R. Vollgraf (2019). Flair : An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 54–59.

Aletras, N., D. Tsarapatsanis, D. Preoţiuc-Pietro, et V. Lampos (2016). Predicting judicial decisions of the european court of human rights : A natural language processing perspective. *PeerJ Computer Science 2*, e93.

Barriere, V. et A. Fouret (2019). May i check again ?–a simple but efficient way to generate and use contextual dictionaries for named entity recognition. application to french legal texts. *arXiv preprint arXiv :1909.03453*.

Chalkidis, I., M. Fergadiotis, P. Malakasiotis, N. Aletras, et I. Androutsopoulos (2020). Legal-bert : The muppets straight out of law school. *arXiv preprint arXiv :2010.02559*.

Devlin, J., M.-W. Chang, K. Lee, et K. Toutanova (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

Douka, S., H. Abdine, M. Vazirgiannis, R. E. Hamdani, et D. R. Amariles (2021). Juribert : A masked-language model adaptation for french legal text. *arXiv preprint arXiv :2110.01485*.

Fernandes, W. P. D., L. J. S. Silva, I. Z. Frajhof, G. d. F. C. F. de Almeida, C. N. Konder, R. B. Nasser, G. R. de Carvalho, S. D. J. Barbosa, H. C. V. Lopes, et al. (2020). Appellate court modifications extraction for portuguese. *Artificial Intelligence and Law 28*(3), 327–360.

Filtz, E., M. Navas-Loro, C. Santos, A. Polleres, et S. Kirrane (2020). Events matter : Extraction of events from court decisions.

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

Honnibal, M. et I. Montani (2017). spaCy 2 : Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Huang, Z., W. Xu, et K. Yu (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv :1508.01991*.

Joulin, A., E. Grave, P. Bojanowski, et T. Mikolov (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv :1607.01759*.

Kjeldgaard, L. et L. Nielsen (2021). Nerda. GitHub.

Konkol, M. et M. Konopík (2015). Segment representations in named entity recognition. In *International Conference on Text, Speech, and Dialogue*, pp. 61–70. Springer.

Lafferty, J., A. McCallum, et F. C. Pereira (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data.

Le, H., L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, et D. Schwab (2019). Flaubert : Unsupervised language model pre-training for french. *arXiv preprint arXiv :1912.05372*.

Li, J., A. Sun, J. Han, et C. Li (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering 34*(1), 50–70.

Lopez, P. (2018). Deep learning framework for text, https ://github.com/kermitt2/delft.

Martin, L., B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de La Clergerie, D. Seddah, et B. Sagot (2019). Camembert : a tasty french language model. *arXiv preprint arXiv :1911.03894*.

Mathis, B. (2021). Extracting proceedings information and legal references from court decisions with machine-learning. *Available at SSRN 3919849*.

McCallum, A., D. Freitag, et F. C. Pereira (2000). Maximum entropy markov models for information extraction and segmentation. In *Icml*, Volume 17, pp. 591–598.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, et J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

Ngompé, G. T., S. Harispe, G. Zambrano, J. Montmain, et S. Mussard (2019). Detecting sections and entities in court decisions using hmm and crf graphical models. In *Advances in Knowledge Discovery and Management*, pp. 61–86. Springer.

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

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

Shankar, A. et V. Buddarapu (2018). Deep ensemble learning for legal query understanding. In *Proceedings of the CIKM 2018 Workshops - International Workshop on Legal DAta Mining (LeDAM 2018)*, Online.

Suárez, P. J. O., Y. Dupont, B. Muller, L. Romary, et B. Sagot (2020). Establishing a new state-of-the-art for french named entity recognition. *arXiv preprint arXiv :2005.13236*.

Trias, F., H. Wang, S. Jaume, et S. Idreos (2021). Named entity recognition in historic legal text : A transformer and state machine ensemble method. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pp. 172–179.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, et A. M. Rush (2020). Transformers : State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, Online, pp. 38–45. Association for Computational Linguistics.

Remerciements

Cette recherche a bénéficié du support de la région Occitanie pour l'allocation doctorale de S.A. Mahmoudi. Les auteurs remercient l'Agence nationale de la recherche, pour le financement du projet LAWBOT ANR-20-CE38-0013 : Apprentissage profond pour la modélisation prédictive de la jurisprudence.

Summary

In this paper we study named entities recognition in court decisions using a CamemBERT trained on legal documents. We compare its performance to BiLSTM-CRF and BiLSTM-CRF ensemble models. We improve the performance of the basic CRF model Ngompé et al. (2019).