Universität Heidelberg

Institut für Computerlinguistik
Seminar: Semantic Role Labeling

Performanz verschiedener Embeddings auf dem *state-of-the-art* spannen-basierten Modell zum Semantic Role Labeling

Dang Hoang Dung Nguyen
Matrikelnummer: 3512492
nguyen@cl.uni-heidelberg.de
Sommersemester 2019

Dozentin: Éva Mújdricza-Maydt

20. April 2020

Inhaltsverzeichnis

1 Einführung **2**

2 Das Modell und die Embeddingstypen **3**
2.1 Das Modell Ouchis (2018) 3
2.2 Embeddingsmodelle 4
2.2.1 Typisches nicht-kontextbasiertes Wortembedding 4
2.2.2 Kontextualisierte Wortembeddings 4

3 Durchführung der Experimenten **5**
3.1 OntoNotes CoNLL 2012 5
3.2 Setup 6
3.2.1 Embeddings 6
3.2.2 Parameter und Hyperparameter 7
3.3 Ergebnisse 7

4 Analysen **8**
4.1 Performanz in Spannen- und Labelerkennung 8
4.2 Label Verwechselung 9
4.2.1 Konfusionsmatrix für Labelingsfehler 10
4.2.2 Die erlernten Label-Embeddings 11

5 Diskussion **12**
5.1 Überanpassungsproblem 12
5.2 Kontextualisierte Embeddings mit diesem Modell 13

6 Verwandten Arbeiten **14**

7 Fazit **14**

8 Appendix **15**

Kurzfassung

Ziel des Tasks *Semantic Role Labeling* ist die Entdeckung der Struktur von Prädikat-Argumenten eines Satzes. Aktuell gibt es zahlreiche Modelle, die effizient diesen Task gelöst und relativ positive Ergebnisse geliefert haben. Eines davon ist das spannen-basierten Modell von Ouchi et al. 2018. Diese Ausarbeitung repräsentiert die Experimenten auf diesem Modell mit verschiedenen Embeddingstypen, nämlich SENNA, ELMo, BERT und Stacked Flair-BERT, die gute Performanz auf anderen Tasks wie NER, POS-Tagging, usw... hatten. Diese Experimenten erzielen die mögliche Lösung bzw. Verbesserung der Schwachheit des Modells, hauptsächlich das Verwechseln zwischen trainierten Embeddings der Argumenten. Ergebnisse haben gezeigt, dass BERT und Stacked Flair-BERT Embeddings besser als ELMo (*state-of-the-art*) performiert haben. Außerdem lässt es sich die Frage offen, ob die Auswahl von geeigneten Hyperparametern (in diesen Experimenten hauptsächlich mit Epochenanzahl) die Ergebnisse verbessern oder verschlechtern könnte.

1 Einführung

Semantic Role Labeling (SRL) beschäftigt sich mit der Erkennung von Argumenten zu(m) Prädikat(en) eines Satzes. Dieser Task beinhaltet vier Teilaufgaben zur Erkennung von drei Komponenten: Prädikaten, Wortphrasen, die Argumenten davon zuweisen, und ihre zugeordneten semantischen Rollen. In dieser Arbeit ist SRL nach *PropBank* in zwei Formen eingeteilt. Diese zwei repräsentieren zwei Annotationsarten der semantischen Argumenten. Eine ist auf Dependenz basiert, d.h. der syntaktische Kopf jeder Wortphrase ist erkannt und mit der entsprechenden semantischen Rolle annotiert. Unter der anderen Art ist jede Wortphrase als eine Spanne zu betrachten. Die Argumenten sind deshalb auf diesen Spannen basiert (Palmer et al., 2005).

Im Mittelpunkt dieser Arbeit ist die Ausarbeitung des spannen-basierten Modells von Ouchi et al. 2018. Die Autoren haben Experimenten für ihr Modell mit zwei Embeddingstypen *SENNA* (Collobert et al., 2011) und *ELMo* (Peters et al., 2018) an zwei Datensätzen *CoNLL-2005* (Carreras and Màrquez, 2005) und *CoNLL-2012* (Pradhan et al., 2012) vorgeführt. In dieser Arbeit wurde das Modell erneut mit den vorgestellten Embeddingsmodellen auf dem Datensatz CoNLL-2012 getestet. Danach folgt dieselbe qualitative Analyse auf den erworbenen Ergebnissen. Ouchi und seine Kollegen (2018) haben festgestellt, dass viele Argumenten oft miteinander durch das Modell verwechselt wurden. Es lässt sich die Frage offen, ob das Umgehen dieser Schwachheit anderen neuen Technik verlangt oder die Anwendung anderer Embeddingsmodelle, die bessere Ergebnisse als *ELMo* im an-

deren Tasks (Wiedemann et al., 2019) ausgegeben haben, das verbessern könnte. In dieser Ausarbeitung sind das *BERT* (Devlin et al., 2019) Embedding und das Stacked Embedding zwischen *Flair* (Akbik et al., 2018) und *BERT* im Modell experimentiert.

Die folgende gliedert sich in vier weiteren Abschnitten. Der nächste Abschnitt gibt ein Kurzüberblick über das spannen-basierte Modell Ouchis und die experimentierten Embeddingstypen. Danach folgen die Vorgehensweise und Ergebnisse auf dem Testset von *CoNLL 2012*. Anschließend analysiere ich die Accuracy in Erkennung der Spannen und Labels und das Verwechseln der erlernten Embeddings der Labels auf letztendlichen Ergebnissen. Abschließend ist eine kleine Diskussion über die unerwarteten Ergebnisse bzw. meine Ansicht zur Auswahl der Format von kontextualisierten Embedding als Eingabe des Modells.

2 Das Modell und die Embeddingstypen

In diesem Kapitel beschreibe ich kurz das Modell Ouchis, und vier verschiedene Embeddings, die ich mit dem Modell trainiert habe. Außer *SENNA* und *ELMo* Embeddings, die erneut nach (Ouchi et al., 2018) experimentiert wurden, wende ich noch zwei neue 2018/2019 vorgestellte kontextualisierte Embeddings an, die relativ häufig parallel mit ELMo in verschiedenen Tasks untersucht werden, *BERT* und *Flair*.

2.1 Das Modell Ouchis (2018)

Das spannen-basierte Modell Ouchis (2018) löst zwei Teilaufgaben des Tasks SRL, nämlich Prädikats- und Spannenerkennung, basierend auf der Argumenten. Durch die aufgelisteten semantischen Rollen, die im Datensatz vorhanden sind, wird jedes der Prädikate eines Satzes und die Spanne, die seine gefundenen Argumente zuweisen, erkannt.

Die Autoren haben ein gestapeltes (oder als *Stacked*) BiLSTM-Spannen-basierten Modell aufgebaut, um alle möglichen Spannen eines Arguments des Prädikats im Satz zu repräsentieren. Von denen berechnen sie die Scores bzgl. der zubetrachtenden Argumenten und selektieren für jedes die Spanne, die das höchste Score hat. Um mehrere sich miteinander überlappenden Spannen von verschiedenen Argumenten als Endergebnisse nicht zu bekommen, entwerfen die Autoren noch einen *Greedy-Search*-Algorithmus (Ouchi et al., 2018).

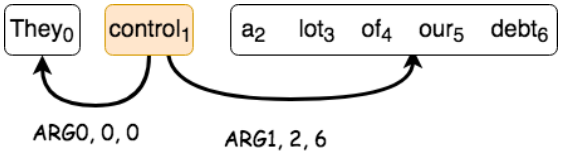

Abbildung 1: Beispiel von ausgegebenen Ergebnissen des Modells

Abbildung 1 stellt ein Beispiel von den Endergebnissen des Modells auf dem *CoNLL 2012* Testset mit *SENNA* Embedding dar. Das Prädikat `control` in der Spanne (1, 1) hat zwei Argumenten in zwei Spannen: Spanne (0, 0) mit der semantischen Rolle `ARG0` ist *they* und Spanne (2, 6) mit der Rolle `ARG1` ist *a lot of our debt*.

Mit dieser Architektur hat dieses Modell eventuell mit Ensemblingstechnik den aktuellen *state-of-the-art* Resultat auf dem *CoNLL 2012* Testset mit F1-Score von 87.0 mit *ELMo* Embedding erreicht. Das einzelne Modell[1] hat auch gute Ergebnisse von 83.0 mit *SENNA* und 86.2 mit *ELMo* Embedding (Ouchi et al., 2018), und übertraf das *BiLSTM-CRF*-Modell (Lafferty et al., 2001; Zhou and Xu, 2015).

2.2 Embeddingsmodelle

Wortembeddings haben häufig ausgezeichnete Performanzen mit neuralen Modellen. Untersucht und experimentiert sind zwei Embeddingstypen: kontextbasierte und nicht-kontextbasierte. Diese beiden standen im Fokus von Ouchis Experimenten, sowohl *ELMo* als auch *SENNA* und haben ganz gute Resultate geliefert, besonders das kontextualisierte Embedding.

2.2.1 Typisches nicht-kontextbasiertes Wortembedding

SENNA (Collobert et al., 2011) ist eines der typischen statischen Wortembeddings, die Kontexte ignorieren. Jedes Wort hat nur eine Repräsentation und seine verschiedene Bedeutungsebenen sind nicht unter Betrachtung.

2.2.2 Kontextualisierte Wortembeddings

Im Gegensatz zu *SENNA* und den ähnlichen, spielen Kontexte eine bedeutende Rolle für kontextualisierten Embeddings, weil sie die semantischen Ebenen des Wortes tragen. D.h. ein Wort kann mehr als eine Repräsentation haben, basierend auf ihren Kontextfenstern, da sie das Problem von Polysemie löst. Es hat sich

[1] Single Model

gezeigt, dass sie die Performanzen in vielen Downstream NLP Tasks wie Sequenztagging, Textklassifikation, maschinelle Übersetzung (Wiedemann et al., 2019) drastisch verbessert. In meinen Experimenten erörtete ich drei Embeddingsmodelle. Neben *ELMo*, die schon von Ouchi vorab überprüft sind, werden zwei anderen Embeddings aus *BERT* und *Flair* experimentell untersucht. Diese drei sind durch drei verschiedene Architektur und neuronale Modelle aufgebaut.

- **ELMo**: *Embedding from Language Model* (Peters et al., 2018). In diesem Modell für jedes Wort sind in jedem Hidden Layer zwei gestapelten RNN[2], einmal vorwärts, einmal rückwärts im großen ungelabelten Korpus trainiert. Ausgabe jedes Layers ist der Summenvektor oder Konkatenation der beiden Netzwerke.

- **Flair**: (Akbik et al., 2018) Dieses Embedding verwendet dasselbe Architektur wie *ELMo*, auch vorwärts-rückwärts RNNs. Der Unterschied liegt an der Einheit der trainierten Netzwerke, nämlich Charakter, statt Wort. Ausgabe jedes Layers ist die Konkatenation der Ausgabenvektoren von Vorwärts-Modell und Rückwärts-Modell. Flair-Embedding ist eher als Sequenzembedding zu untersuchen, und Sequenz repräsentiert nicht nur Token(s) sondern auch Sätze.

- **BERT**: *Bidirectional Encoder Representation from Transformer* (Devlin et al., 2019) Zum Trainieren dieses Embedding ist eine andere Architektur aufgebaut. Anstatt jedes Wortes auf zwei uni-direktionalen Modellen zu trainieren, bauten Devlin und seine Kollegen ein Self-Attention Transformer Modell, das Informationen von beiden Kontextfenstern links und rechts des Wortes gleichzeitig erfasst. Dieses Modell entschärft die Uni-direktionalität von zwei vorherigen Modellen durch seine Kombination mit *Masked Language Model* (MLM) vor dem Training. Dieses blendet manchen[3] randomisierten Tokens im Input aus und ersetzt sie durch `[MASK]`, zwecks Identifizierung ihrer vorgestellten IDs im gesamten Vokabular des Korpus beruhend nur auf ihren Kontexten. Außerdem versuchten sie vor dem Training die Beziehung zwei nebeneinander stehenden Sätze zu formulieren, welche kaum von traditionellen bidirektionalen Sprachmodellen erfasst werden. (Young et al., 2017)

3 Durchführung der Experimenten

3.1 OntoNotes CoNLL 2012

Alle Experimenten wurden auf dem Datensatz *ConLL 2012* (Pradhan et al., 2012) durchgeführt. Diese Daten sind spezifisch nach spannen-basierten Repräsentation

[2] Recurrent Neural Network

[3] vordefiniert sind 15% der gesamten Tokens im Satz

von *PropBank*[4] erstellt.

Trainingset	Developmentset	Testset
74943 [5]	9603	9479

Tabelle 1: Statistiken vom *Shared Task Datensatz CoNLL 2012* (Anzahl der Sätze)

3.2 Setup

3.2.1 Embeddings

Ouchi et al. haben in ihren Code zwei Varianten angeboten, eines für nicht-kontextualisierten Embedding wie *SENNA*, das andere für kontextualisierten Embeddings wie *ELMo*. Es ist bei kontextualisierten Embeddings strikt definiert, dass Embeddingsinput drei Layers enthält. Mit d^{word}[6] Features, ist jedes Wort im Datensatz durch einen (3, d^{word})-Vektor repräsentiert. Insgesamt sind im Folgenden 4 Embeddings, die aus 4 vor-trainierten vorgestellten Embeddings erlernt wurden, experimentiert. Die Vorgehensweise ist wie folgt:

- *SENNA* Embedding habe ich von `http://ronan.collobert.com/senna/` übernommen.
- Aus den vor-trainierten Modellen[7] im ***ALLENNLP*** trainierte ich die Datensätze und speicherte *ELMo* Embeddings der letzten drei Hidden Layers.
- *BERT* Embeddings für die Datensätze sind auf dem Korpus *kleingeschriebenen englischen Texten* mit dem neu entwickelten Technik *Whole Word Masking* trainiert. *Whole Word Masking*[8] strebt auf Subworte oder Worte an, die aus mehreren Tokens bestehen. Mit diesem Technik, neben zufälliger Ausblendung der Tokens (nach vordefinierten Rate) im Satz, sucht das Modell explizit die (möglicherweise nebeneinander stehenden) Tokens aus, die zusammen miteinander ein Wort bilden, und ersetzen sie auch durch `[MASK]`.

  Beispiel wie folgt:

[4] `https://propbank.github.io/`
[5] exklusiv der Sätze im Dokument `wb/eng/00/eng_0003` wegen fehlender Annotation der semantischen Rollen
[6] alle erwähnten d^{word} hier sind 1024 gemeint
[7] `https://github.com/allenai/allennlp/`
[8] `https://github.com/google-research/bert/#bert`

Input Text: the man jumped up , put his basket on phil ##am ##mon ' s head
Original Masked Input: `[MASK]` man `[MASK]` up , put his `[MASK]` on phil `[MASK]` ##mon ' s head
Whole Word Masked Input: the man `[MASK]` up , put his basket on `[MASK]` `[MASK]` `[MASK]` ' s head

Analog zu *ELMo*, speicherte ich auch die Vektoren der letzten drei Layers jedes Wortes.

- Das letzte Embedding ist ein gestapeltes Embedding (im weiteren kann auch als *Stacked Embedding* genannt werden) bestehend aus *Flair* (Akbik et al., 2018, 2019)[9] und *BERT* Embeddings. *Flair* Embedding ist auf Charakter-Level vortrainiert und eher als Sequenzembedding untersucht. In ihren Experimenten schlagten Akbik und seine Kollegen vor, dieses Embeddings mit den anderen zu konkatenieren. In meinem Experiment, für jedes Wort, konkatenierte ich es mit meinem oben stehenden BERT Embedding[10]. Aus diesem konkatenierten (1, $3 \times d^{word}$)-Embedding bildete ich zum (3, d^{word})-Vektor um. Dieses stellt ein Wort im Datensatz dar. Mein Embedding in diesem Fall hat nicht mehr dasselbe Merkmal wie die anderen, dass jeder (1, d^{word})-Vektor in dem (3, d^{word})-Vektor ein Layer des Embeddings entspricht, sondern jeweils *Vorwärts-Flair-*, *Rückwärts-Flair* Embedding[11], *BERT* Embedding. Motivation zu diesem Aufbau kläre ich in weiteren Abschnitten auf.

3.2.2 Parameter und Hyperparameter

Alle anderen Parameter und Hyperparameter folgen das originelle Setup in (Ouchi et al., 2018)[12].

3.3 Ergebnisse

Die folgende Tabelle stellt die neu experimentierten Endergebnisse dar (nach einem Durchlauf)[13].

[9] `https://github.com/zalandoresearch/flair`

[10] anstatt die drei Layers zu speichern, berechnete ich ein durchschnittlichen Vektor der Layers, damit ich für jedes Wort ein (1, d^{word})-Vektor bekam

[11] beide Embeddings sind auf dem Korpus der englischen Nachrichten von 1 Mrd. Worte trainiert, CPU-freundlich

[12] zusätzliche Skripts zur Bearbeitung ist hier zu finden `https://github.com/jasmine95dn/srl-context-emb`

[13] alle unten stehenden Ergebnisse sind von meinen Experimenten gesammelt, d.h. das kann variieren bzw. nicht gleich wie in originellen Experimenten sein, da ein Dokument (siehe 5) ausgelassen war

		Development			Test		
EMB	**Zeitpunkt**	**P**	**R**	**F1**	**P**	**R**	**F1**
SENNA	Traningsende	83.2	79.1	81.0	83.8	79.6	81.6
ELMo	Trainingsende	86.0	83.3	84.6	6.6	15.6	8.8
BERT	Trainingsende	86.3	**84.9**	**85.6**	**86.5**	**85.1**	**85.8**
Flair-BERT	Traningsende	**87.0**	84.2	85.5	3.2	9.3	4.7
	Epoch 19[14]	85.8	83.0	84.4	86.1	83.4	84.8

Tabelle 2: Experimentelle Ergebnisse auf dem Datensatz CoNLL 2012 (im Sep.-Okt./2019 experimentiert und gesammelt)

Durch diese Experimenten ist es festzulegen, dass *BERT*-Embedding außer Precision im Developmentset die beste Ergebnisse sowohl im Development- als auch im Testset hatten. Mit *BERT*-Embedding ist deutlich zu erwarten, dass es noch besser in Ensemble Modell wird, und wahrscheinlich zukünftig neue *state-of-the-art*-Resultate im Task SRL erreichen könnte.

Noch zu merken ist das Ergebnis von *ELMo* und *Stacked Flair-BERT* auf dem Testset viel niedriger als die Scores auf dem Developmentset. Deswegen habe ich, für das *Stacked Embedding BERT-Flair*, den zweiten Trainingsdurchlauf ausgeführt und nach jeder Epoche auf dem Testset getestet um zu finden in welchem Zeitpunkt das Ergebnis auf dem Testset stark abweicht, welches ein Anfang der Überanpassung[15] kennzeichnet. Dieser Zeitpunkt ist überraschend gleich nach Epoch **19**. Dieses Resultat ist in Tabelle 2 zu sehen. Seine Scores auf dem Testset sind besser als die mit *SENNA* und seine Scores auf dem Developmentset fast gleich wie die mit *ELMo*. Dieses *Stacked Embedding* kann auch, neben *BERT*-Embedding, ein potenziell Wortembedding für das neue *state-of-the-art*-Ergebnis erreichen, wenn alle anderen Hypeparameter (in diesem Fall, Anzahl der Epochen) gut kontrolliert werden und spezifische Techniken, die mit Überanpassung umgehen, extra zum Modell angewandt werden.

4 Analysen

4.1 Performanz in Spannen- und Labelerkennung

Wegen unerwarteter Ergebnisse von *ELMo* und *Stacked Flair-BERT* voraussichtlich aufgrund Overfittings und da ich die beste Resultate für diese Analyse erziele,

[13] Early Stopping: ein Typ von Regularization
[15] Overfitting, wird noch wieder im **5.1** diskutiert

nehme ich für *Stacked Flair-BERT* das Testergebnis gleich vor dem Zeitpunkt vom Overfitting. Tabelle 3 und 4 drücken die Resultate aus.

EMB	**F1**
SENNA	86.0
ELMo	16.9
BERT	90.2
Stacked Flair-BERT	89.1
SENNA (CRF Modell)	87.9
ELMo (CRF Modell)	**90.9**

Tabelle 3: F1 Scores für übereinstimmten Spannengrenzen

EMB	**Accuracy**
SENNA	95.0
ELMo	52.4
BERT	**95.1**
Stacked Flair-BERT	95.0
SENNA (CRF Modell)	93.6
ELMo (CRF Modell)	94.4

Tabelle 4: Accuracy für semantische Rollen

Obwohl *BERT* und eben *Stacked Flair-BERT* F1 Scores in der Erkennung der Spannengrenzen mit erhöht haben (90.2), bleibt das *CRF-basierte* Modell ein kräftiges Modell zu dieser Teilaufgabe (siehe Tabelle 3). Da die beiden Embeddings mit *BERT* die F1 Scores des spannen-basierten Modells verbessert haben, scheinen diese Maße des *CRF-basierten Modells* mit *BERT*-Embedding steigen zu können.

In Tabelle 4 ist es deutlich ausgedrückt, dass spannen-basierte Modelle besonders gut in Zuweisung der richten semantischen Rollen einer Spanne bzgl. des Prädikats. Die Ergebnisse sind deshalb vorzusehen, mit Accuracy 95.1 für *BERT*-Embedding.

4.2 Label Verwechselung

Ein wichtiges Problem, das von Ouchi et al. besonders berücksichtigt wurde, ist die Verwechselung zwischen häufigen Labels. Anfangs dieser Ausarbeitung stelle ich mir die Frage, ob es verlangt wird, neues Feature bzw. Technik um dies zu lösen, oder ein kräftigeres kontextualisierteres Embedding als *ELMo* ausreicht. Im Weiteren gehe ich in zwei Kriterien ein, die schon von Ouchi analysiert waren. Da *BERT*-Embedding die beste Ergebnisse geliefert hat (siehe Tabelle 2), steht dieses Embedding im Mittelpunkt der Analysen.

4.2.1 Konfusionsmatrix für Labelingsfehler

Anstatt Ouchis Zählungsweise auf dem Spannen-Level zu folgen, habe ich die Anzahl der vorhergesagten Argumenten auf dem Wort-Level gesammelt. Statt nur die Argumenten, die richtige Spannengrenzen zuweisen, betrachte ich Worte, die zu den richtigen Argumenten gehören. Meiner Ansicht nach sollten diese Ergebnisse nicht viel von den nach Ouchis Methode erhaltenen Resultaten abweichen, da *BERT* schon bessere Ergebnisse in Erkennung der Spannengrenzen (siehe Tabelle 3) hat.

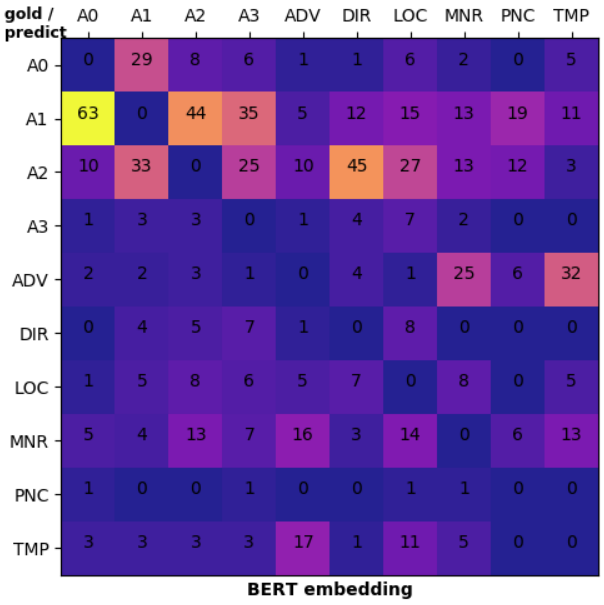

Abbildung 2: Konfusionsmatrix für Labelingsfehler am Trainingsende vom Modell mit
BERT Embedding

In (Ouchi et al., 2018) wurden zwei Hauptfehler genannt, nämlich (1) die Verwechselung zwischen `A0` und `A1` bzgl. transitiver-intransitiver Verben (können als *labile Verben* genannt werden) und (2) dass `A2` ganz häufig mit den Adjunkten `DIR` und `LOC` verwechselt wird, da `A2` für viele Verben die semantische Relation als Richtung oder lokale Angabe trägt.

In Abbildung 2 entsteht das Resultat aus dem Modell mit *BERT*-Embedding. Anscheinend sind die zwei Fehler immer noch mit relativ höhen Werten zu sehen. Dies hat bestätigt, dass die genannten Hauptfehler nicht durch andere mächtigere

Embeddings[16] behoben werden können und die Anwendung von verbalen gefassten Kenntnissen (Ouchi et al., 2018) als eine sinnvolle Lösung zu untersuchen wäre.

4.2.2 Die erlernten Label-Embeddings

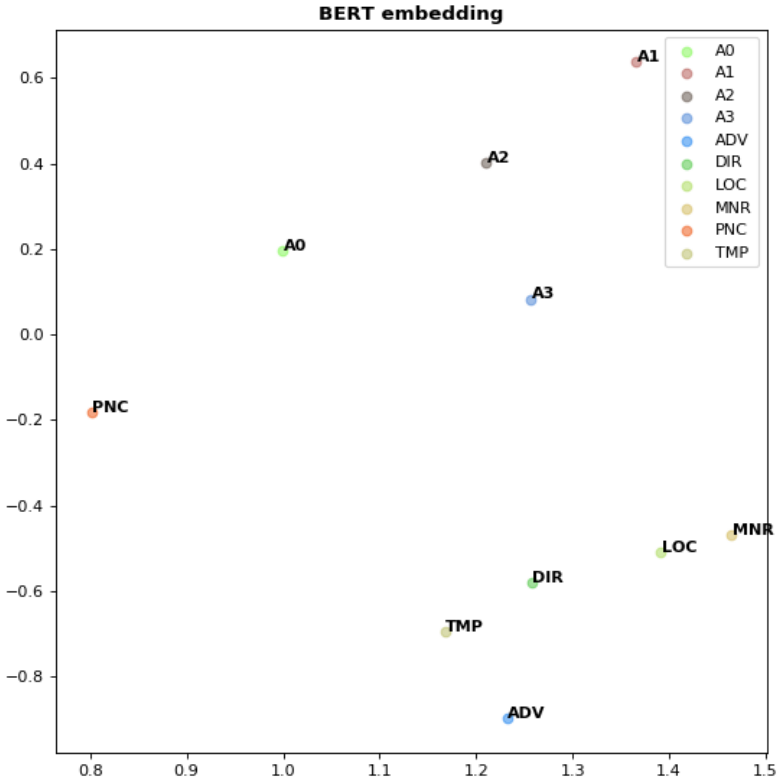

Abbildung 3: Label Embeddings von mit *BERT* trainierten häufigen Labels

Analog zu der gleichen Analyse von Ouchi et al., untersuche ich die Verteilung der erlernten Label-Embeddings. Aus Abbildung 3 kann es gesehen werden, dass alle häufigen Labels nicht mehr nebeneinader stehen. Die vorherigen von Ouchi et al. genannten Probleme, wie dass das Kernargument `A2` nah zum Adjunkt `DIR` steht oder die Adjunkte relativ ein Cluster bilden, is hier gelöst. Das Geschehen, dass die Verwechselungen entstehen, obwohl in der Tat die Accuracy für jedes dieser Labels ziemlich hoch sind (siehe Appendix Abbildung 5) könnte deswegen eine zukunftsträchtige Forschung sein.

[16] im Appendix befindet sich die Konfusionsmatrix vom denselben Typ mit Stacked Flair-BERT Embedding, siehe Abbildung 6

5 Diskussion

5.1 Überanpassungsproblem

Aus Tabelle 2 ist es deutlich gezeigt, dass Überanpassung / Overfitting entstand, wegen der stärker Abweichung zwischen Ergebnissen auf dem Developmentset und Testset (bei *ELMo* und *Stacked Flair-BERT*). Die zwei folgenden Abbildungen verdeutlichen dieses Problem.

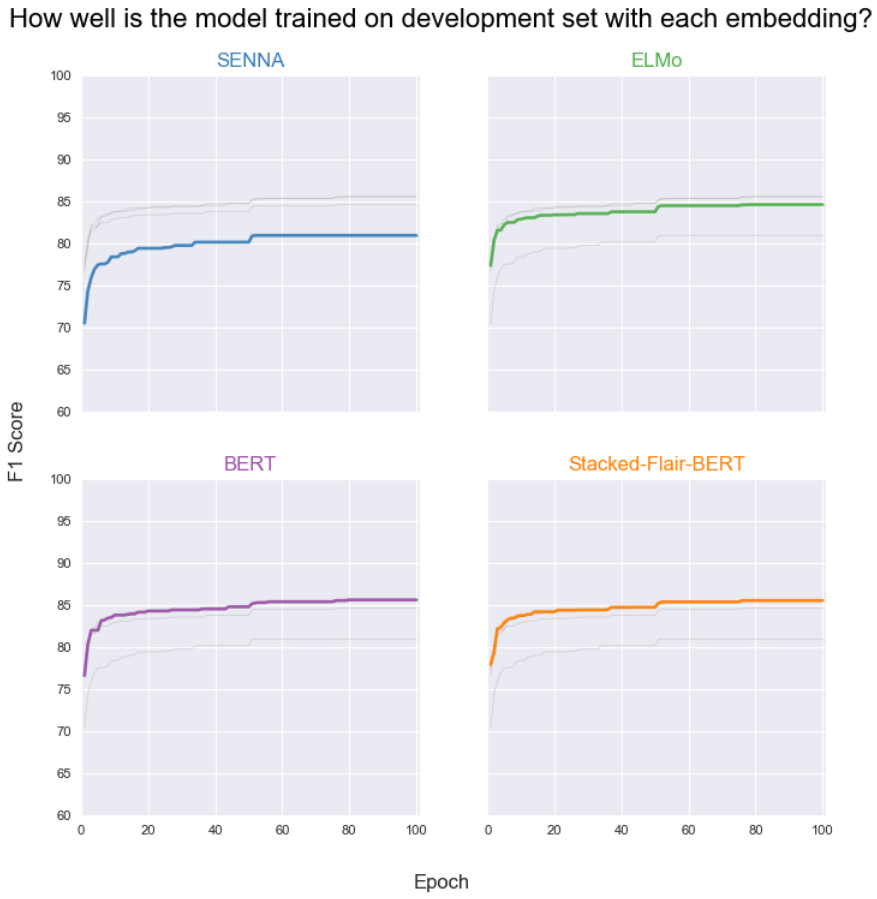

Abbildung 4: Performanz der vier Embeddings auf dem Developmentset vom CoNLL 2012

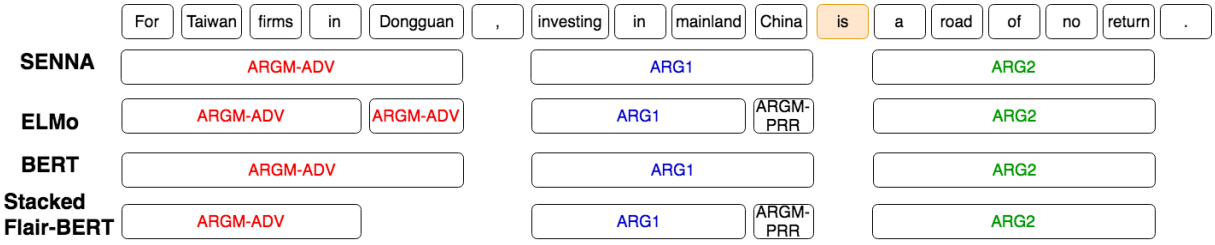

Abbildung 5: Bestimmung von semantischen Rollen und Spannen des Modells auf dem Testset vom CoNLL 2012 mit vier Embeddings am Trainingsende

Aus Abbildung 4 ist es merkwürdig, dass die kontextualisierten Embeddings besser auf dem Developmentset performiert haben, als *SENNA*. Das widerspricht das auf Abbildung 5. Der Beispielsatz ist aus dem Testset von *CoNLL 2012*. Wie *ELMo* und *Stacked Flair-BERT* Spannen erkennen und Argumenten zuweisen, im Vergleich zu *SENNA* und *BERT* drückt stark aus, wie das Modell mit ihnen deutlich während Trainings überangepasst war und dass die Spannen sehr klein wie möglich zerlegt waren, um passende semantische Rollen zu finden. Es führt zu der Vermutung, dass neben zwei häufigen Verwechselungen in **4.2**, das Verwechsel zwischen häufige Labels mit sehr seltenen Labels wie `ARGM-PRR` identifiziert werden kann und dies ist nicht zu erwarten. Ouchi et al. scheinen nicht dieses Problem mit *ELMo* diskutiert zu haben, obwohl sie besonders bei Regularization ziemlich viele Hyperparameter betrachten, um vermutlich Überanpassungsproblem zu überwinden. Die Strategie *Early Stopping* (Raskutti et al., 2011), die ich für das Training von *Stacked Flair-BERT* angewandt habe, hat funktioniert (Tabelle 2). Zukünftig soll dieses Problem behandelt werden, da es keine Garantie gibt, dass Training des Modells mit weiteren neuen Embeddings, die aus ähnlichen neuronalen Modellen wie *ELMo* und *Flair* aufgebaut sind (siehe **2.2.2**), nicht zu derselben Situation kommen. Hier bleibt die Frage offen, ob Training des Modells mit aus Transformer aufgebauten Embedding wie *BERT* Overfitting vermeiden kann.

5.2 Kontextualisierte Embeddings mit diesem Modell

Als Eingabe für Training des Modells mit *ELMo* ist es verlangt (dies ist bei dem Durchlauf festgestellt), dass jedes Wort durch ein (3, d^{word})-Vektor repräsentiert wird, anstatt (1, d^{word})-Vektor. Dies stellt mir die Frage, ob die Annahme von viel mehr (letzten) Layers eines gleichen vor-trainierten Modells eine deutliche Rolle im Training spielt, da jedes Layer ähnliche Merkmale hat, weil sie aus einem Modell stammen. Deswegen habe ich das Stacked Embedding von *Flair* und *BERT* (siehe **3.2.1**) aufgebaut. Was dieses Embedding sich von den anderen in Experimenten unterscheidet, liegt genau bei den Layers, nämlich jedes Layer hat Merkmale eines

unterschiedlichen neuronalen Modells. Tabelle 2 und Abbildung 4 haben gezeigt, wie mächtig mein *Stacked Embedding* ist. Das Modell mit ihm entwickelt genau so stark wie das mit *BERT*. Das mit *BERT* braucht 80 Epochen (Ergebnisse aus Experimenten, siehe 12) und dass das Lernengrad zweimal nach 25 Epochen halbiert ist, um das Ergebnis im Testset zu erreichen, wobei das mit *Stacked Flair-BERT* nur nach 20 Epochen schon ein ziemlich überraschendes Ergebnis geschafft hat und gleich zu Overfitting kam. Neben Merkmalen von *Flair* in zwei Layers hat dieses Modell in einem das Merkmal von durchschnittlichen auf 3 Layers verteilte *BERT*-Embedding. Die Mächtigkeit dieses Embeddings kann daraus begründet werden. In der Zukunft erhoffe ich eine ausführliche Untersuchung dieses Stacked Embeddings (*Flair-BERT*) in weiteren SRL Modelle und möglicherweise in anderen Tasks.

6 Verwandten Arbeiten

SRL Modelle mit BERT: Nach (Ouchi et al., 2018) gab es ziemlich viele neue SRL Modelle zwecks Erreichens neues *state-of-the-art*. Zwei davon haben *BERT*-Modell in ihrem Modell verwendet und relativ positive Ergebnisse erreicht, obwohl sie das Ensemble-Modell Ouchis nicht übertroffen haben. Eines davon, (Shi and Lin, 2019), haben direkt ihr Modell aus *BERT* (Devlin et al., 2019) aufgebaut, und *BERT*-Embedding als eine der Eingaben. Ihre beste Ergebnisse haben aber die Ergebnisse des einzelnen Modells von Ouchi et al. auf beiden SRL Datensatz *CoNLL 2012* (Pradhan et al., 2012) und *ConLL 2005* (Carreras and Màrquez, 2005) übertroffen, und waren nur 0.3% weniger auf dem *CoNLL 2005* und 0.6% weniger auf dem *CoNLL 2012* im Vergleich zu dem Ensemble-Modell Ouchis. Dieses Modell hat gezeigt, wie gut *BERT*-Embedding auf verschiedenen Modellen für den Task SRL performiert hat.

Vergleich von verschiedenen kontextualisierten Wortembeddings ELMo, BERT, und Flair: Ein neu veröffentliches Paper, das Performanz dieser drei kontextbasierten Embeddings untersucht hat, ist von (Wiedemann et al., 2019) um semantische Merkmale für Task *Word Sense Disambiguation* zu testen. Sie haben auch dasselbe Befunden, dass sie mit *BERT* hervorragendes Ergebnis bekommen haben. Und ihr experimentiert *Stacked Embedding* ist aus *Flair* und *GloVe* (Pennington et al., 2014) aufgebaut.

7 Fazit

In dieser Ausarbeitung habe ich vier Embeddings (*SENNA*, *ELMo*, *BERT* und *Stacked Flair-BERT*) auf dem spannen-basierten Modell von Ouchi et al. experimentiert. Mein Ziel am Anfang ist zu testen, ob das Verwechsel zwischen häufi-

gen Labels durch Anwendung der scheinbar mächtigen kontextualisierten Embeddings gelöst werden kann. Durch die Experimenten und Analysen hat es gezeigt, dass der Einsatz der anderen Techniken, beispielweise verbale gefasste Informationen, benötigt ist. Außerdem hat es durch Experimenten gezeigt, dass *BERT*-Embedding potentiell neue *state-of-the-art* erreichen kann, wenn es auf Ensemble-Modell Ouchis trainiert wird. Und das *Stacked Embedding Flair-BERT* mit ihren versteckten Fähigkeiten ist möglicherweise weiter zu forschen. Das Modell von Ouchi hat das Überanpassungsproblem während Training mit den aus *RNN*-Modell aufgebauten Embeddings betroffen. In weiteren Analysen bzw. Experimenten sollte dieses behandelt werden, da ihre eingestellte Hyperparameter für Regularization nicht effizient für solches funktioniert hat. Die Kontrolle auf der Trainingsdauer (nämlich die Anzahl der Epochen) wäre eine Variante dazu.

8 Appendix

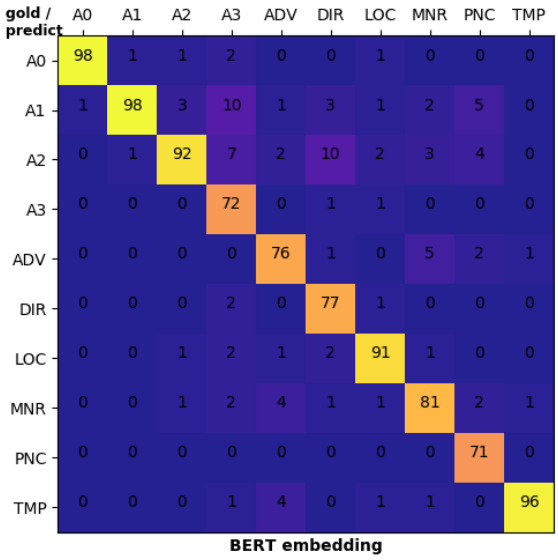

Abbildung 6: Konfusionsmatrix für richtige Labeling am Trainingsende vom BERT Embedding

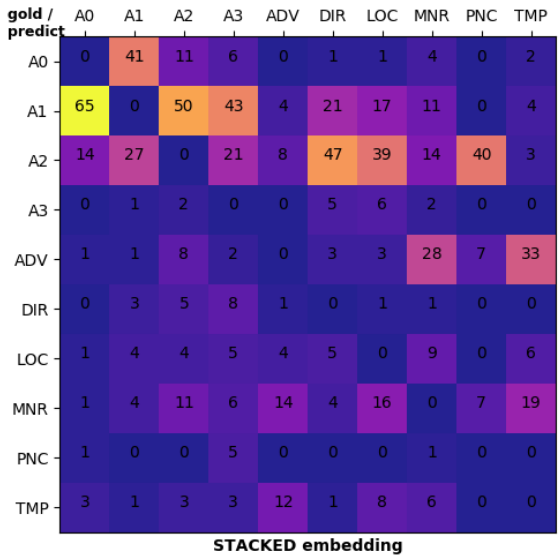

Abbildung 7: Konfusionsmatrix für richtige Labeling am Trainingsende vom BERT Embedding

Literatur

Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C18-1139`.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4010. URL `https://www.aclweb.org/anthology/N19-4010`.

Xavier Carreras and Lluís Màrquez. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164. Association for Computational Linguistics, 2005. URL `http://www.aclweb.org/anthology/W05-0620`.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Processing (Almost) from Scratch

. *Journal of Machine Learning Research (JMLR)*, 12:2493–2537, 2011. URL http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. URL https://arxiv.org/abs/1810.04805.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc., 2001. URL http://www.cs.cmu.edu/afs/cs/usr/lafferty/www/pub/crf.ps.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/D18-1191.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106, March 2005. ISSN 0891-2017. doi: 10.1162/0891201053630264. URL http://dx.doi.org/10.1162/0891201053630264.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237, 2018. URL http://www.aclweb.org/anthology/N18-1202.pdf.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, CoNLL '12, pages 1–40, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2391181.2391183.

G. Raskutti, M. J. Wainwright, and B. Yu. Early stopping for non-parametric regression: An optimal data-dependent stopping rule. In *2011 49th Annu-*

al Allerton Conference on Communication, Control, and Computing (Allerton), pages 1318–1325, Sep. 2011. doi: 10.1109/Allerton.2011.6120320. URL https://ieeexplore.ieee.org/document/6120320.

Peng Shi and Jimmy Lin. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255, 2019. URL http://arxiv.org/abs/1904.05255.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Christian Biemann. Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. *ArXiv*, abs/1909.10430, 2019. URL https://arxiv.org/abs/1810.04805.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13:55–75, 2017. URL https://arxiv.org/abs/1708.02709.

Jie Zhou and Wei Xu. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137. Association for Computational Linguistics, 2015. URL http://www.aclweb.org/anthology/P15-1109.