الجمهورية العربية السورية

وزارة التعليم العالي

جامعة تشرين– كلية الهندسة المعلوماتية

قسم الذكاء الصنعي

تسمية التسلسل اعتمادًا على نموذج جَمعي وتكديس التضمينات لمهمة استخراج الجانب

Stacked Embeddings and Ensemble Model-based Sequence Labeling for Aspect Extraction

مشروع تخرج

إعداد الطالب

علي حيدر أمين أحمد

إشراف

م. ندى جنيدي

العام الدراسي 2021–2022

ملخص

المفتاح الرئيسية لمهمة تحليل مشاعر الجوانب (أهداف الرأي) لمراجعات المنتجات، هو استخراج جوانب أو ميزات المنتج التي أعرب المستخدمون عن آرائهم بشأنها. يُركّز هذا العمل على استخراج الجوانب اعتمادًا على منهج خاضع للإشراف وباستخدام خوارزميات التعليم العميق. اقترح في هذا العمل نموذج جمعي لاستخراج الجانب يستخدم أربعة أنواع من التضمينات المدربة مسبقًا: اثنان منهم تضمينات عامة والآخَران خاصان بالمجال. اقترح أيضًا استخدام مخطط التمثيل OI لأول مرة. دون استخدام أي إشراف إضافي يحقق هذا النموذج نتائج جيدة جدًا، متفوقًا على أحدث الأساليب الحالية SOTA. هذا أول عمل يقدم تقريرًا عن نموذج جمعي Ensemble Model لاستخراج الجانب يعتمد على تكديس تضمينات سياقية وغير سياقية ويستخدم مخطط التمثيل IO ويحقق أداءًا متفوقًا على مجموعتي بيانات معياريتين SE14-16-Restaurants كما أنه يتفوق على العديد من النماذج في مجموعة البيانات المعيارية SE14-Labtop وعلى خطوط أساس قوية Baselines.

Abstract

One key task of fine-grained sentiment analysis of product reviews is to extract product aspects or features that users have expressed opinions on. This work focuses on supervised aspect extraction using deep learning. This work proposes a novel ensemble model employing three types of pre-trained embeddings for aspect extraction: two general-purpose embeddings and one domain-specific embeddings. It also proposes a novel approach to tagging aspects IO. Without using any additional supervision, this model achieves surprisingly good results, outperforming state-of-the-art sophisticated existing methods. To my knowledge, this work is the first to report such stacked contextualized and non-contextualized embeddings based ensemble model for aspect extraction, a new approach to tagging aspects and achieve state-of-the-art performance on two standard datasets, SE14-16-Restaurants, it also outperforms many models in the SE14-Labtop dataset and on strong baselines.

الفهرس

قائمة الأشكال

قائمة الجداول

قائمة المصطلحات

المصطلح	توضيح
SE-EM	اختصار لاسم النموذج المقترح Stacked Embeddings and Ensemble Model
Ensemble Model	نموذج يجمع بين نماذج مختلفة
State-of-The-Art SOTA	أداء متطور مُصطلح يُشير إلى النماذج التي تحقق أداءًا يتفوق على أداء أحدث النماذج الحالية. في مهام التصنيف يكون اعتمادًا على معيارين أساسيين هما Accuracy F1-score
Hyper parameters	المعاملات العليا للنموذج مثل معدل التعلم وعدد الطبقات والخلايا..إلخ
Long-Short-Term-Memory LSTMs	ذات الذاكرة طويلة- الشبكات العصبية المتكررة قصيرة المدى شبكة عصبية شهيرة الاستخدام مع التسلسل
Conditional Random Fields CRFs	حقول ماركوف العشوائية الشرطية هو نموذج تمييزي احتمالي شهير الاستخدام مع مهام التنبؤ المهيكلة

SkipGram & CBOW	معماريتين لنموذجين مختلفين قدمهما ميكلوف ورفاقه سنة 2016 لتدريب فضاء تضمينات
One-Hot Encoding	طريقة لتمثيل النص، خط الترميز الأحادي
Bag of Words	طريقة لتمثيل النص، حقيبة الكلمات
Term Frequency–Inverse Document Frequency TF-IDF	تكرار المصطلح–تكرار المستند العكسي، طريقة لتمثيل النص.
Epoch	تطبيق عملية انتشار أمامي وخلفي على كامل عينات التدريب
Downstream tasks	المهام النهائية (تحليل المشاعر، استخراج الجوانب، التعرف على الكيان المسمى، الترجمة ..إلخ.)
Named Entity Recognitions NER	التعرف على الكيان المسمى
Structured prediction	مهام التنبؤ المهيكلة أي التي تتبع هيكلاً محددًا في عملية التوقع (استخراج الجانب، التعرف على الكيان ..إلخ.)
Handcrafted features	الميزات (السمات) المصنوعة يدويًا ميزات يتم استخراجها من البيانات اعتمادًا على حدث المبرمج ودون تدخل الآلة
Baselines	خط أساس نموذج أولي بسيط وسريع
Neural architecture search problem NAS	البحث عن العمارة العصبية هو مهمة البحث تلقائيًا عن معمارية واحدة أو أكثر لشبكة عصبية والتي ستنتج نماذج ذات نتائج جيدة

Morpheme	هو أقل لفظة تفيد معنى معينا متضمنا فيها
Masked Language Model MLM	التنبؤ بالكلمات المخفية إحدى الأساليب التي تم تدريب نموذج بيرت عليها
Next Sentence Prediction NSP	التنبؤ بالكلمة التالية إحدى الأساليب التي تم تدريب نموذج بيرت عليها
Fine-tuning	ضبط دقيق في مهام نقل التعلم المقصود هو جعل نموذج ما يُلائم مهمة نهائية. أما في الحالة العامة فهو يعني ضبط المعاملات العليا للنموذج بشكل دقيق.
Syntactic rules-based extraction	استخراج الجانب اعتمادًا على قواعد نحوية
Sequence labeling	تسمية التسلسل أي مثلاً استخراج الجانب أو الكيان المسمى، حيث يكون لدينا تسلسل من الكلمات وعلينا تصنيفها (تسميتها)
Bag-of-N-Grams	طريقة لتمثيل النص. حقيبة من ن-جرام

1- مراجعة وتحضير

1-1- مقدمة

يُعتبر استخراج الجانب أهم خطوة في مهمة تحليل مشاعر الجوانب [2] ولها العديد من التطبيقات [3] ويهدف إلى استخراج أهداف الرأي من نص الرأي (المراجعة). حديثًا، حققت نماذج التعلم العميق الخاضعة للإشراف أداءًا متطورًا [13,48,15,12,1,14]. جميع تلك النماذج يعتمد على الترميز التقليدي المستوحى من مهام NER والمتمثّل في تنسيقات IOB، على الرغم من أن هذا التنسيق يُعتبر قياسي وكان مناسبًا في التعامل مع مهام التعرّف على الكيان المُسمى NER، إلا أنه يجعل المهمة أصعب على نماذج استخراج الجانب والتي تُعتبر مهمة أكثر تعقيدًا بدورها. من جانب آخر معظم تلك النماذج يستخدم الميزات المصنوعة يدويًّا Handcrafted features والمعاجم Lexicons وشبكات عصبية معقدة [48,15,12,13,14] على الرغم من أن هذه الأساليب تحقق أداءًا جيدًا مقارنة بالأعمال السابقة، إلا أن هناك اعتبارين مهمين أيضًا. (1) يُفضّل أن يكون تعلم الميزات آليًّا؛ إن كيفية تحقيق أداء تنافسي دون صياغة الميزات يدويًا هو سؤال مهم. (2) وفقًا لمبدأ نصل أوكام أو شفرة أوكام [16]؛ يُفضَل دومًا النموذج البسيط على النموذج المعقد. لذا فإن تحقيق أداء تنافسي مع إبقاء النموذج بسيطاً قدر الإمكان هو أمر هام. العمل الذي اقترحه يُقدّم هكذا نموذج. اقترح لمعالجة الاعتبار الأول آلية تضمين تعتمد على تكديس أربعة أنواع مختلفة من التضمينات ويبدو أنها تلعب دورًا هامًا جدًا في استخراج الجانب. تُرمّز التضمينات المعلومات المتعلقة بكل كلمة، وجودة التضمينات تحدد كم سيكون سهلًا على الطبقات التالية (LSTM،CNN، Attination ..إلخ) فك ترميز هذه الكلمات لاستخراج معلومات مفيدة. حسّنت التطورات الأخيرة في التضمينات المدربة مسبقًا أداء مهام التنبؤ المهيكلة Structured prediction؛ مناهج تعتمد على التضمينات السياقية مثل Flair و BERT و ELMo و XLM-R [17,19,18,20] أدت إلى تطور كبير في مهام التنبؤ المهيكلة. وجد الباحثين [21] أن تمثيل الكلمة المعتمد على وصل (تكديس) عدة تضمينات سياقية مدربة مسبقًا مع تضمينات غير سياقية مثل Word2vec [22] مع تضمينات محرفيّة [23] يمكنها أن تحسّن الأداء. هذه التضمينات تُدرّب على بلايين الكلمات الموجودة عبر الانترنت وتسمى تضمينات عامة. عادةً ما يعاد تدريب هذه النماذج من البداية أو يُعاد

ضبطها Fine-tuning عند التعامل مع المهام النهائية downstream tasks لتصبح خاصة بالمجال. تُظهر الأعمال السابقة أن استخدام التضمينات الخاصة بالمجال إلى جانب التضمينات العامة يمكنها أن تحسّن الأداء [21,1]. استخدم في هذا العمل كل من التضمينات العامة والتضمينات الخاصة بالمجال، ويُترك للشبكة العميقة أن تقرر أيًّ من التضمينات لديها معلومات أكثر فائدة. لمعالجة الاعتبار الثاني اقترح أسلوبًا جمعيًّا يمزج بين المعلومات المستخرجة من نموذجين BiLSTM و CNN-1D (كلاهما صافٍ) خطيًّا، مع شبكة عصبية متصلة بالكامل Fuly connected وطبقة خرج CRF. اقترح أيضًا وبشكل منفصل نموذجًا اسميه Professor مهمته تحسين أداء النموذج الأساسي.

هذا العمل هو أول عمل يقترح هذه البنية ويعتمد على تضمينات سياقية عامة وخاصة بالمجال وغير سياقية خاصة بالمجال مع طريقة جديدة لتمثيل الجوانب تلعب دورًا مهمًا في تحسين الأداء.

1-2- الدراسات المرجعية

قبل بدء هذا العمل وخلاله، اعتمدت على قراءة العديد من الأوراق البحثية المتعلقة بشكل أو بآخر بالمهمة التي أعمل عليها. ما يلي هو تلخيص لأهم هذه الأوراق:

- Automated Concatenation of Embeddings for Structured Prediction (ACE).
  Authors: Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, Kewei Tu.

  Date: 1 Jun 2021

تعتبر التضمينات السياقية المدربة مسبقًا أفضل طرق تمثيل الكلمات لمهام التنبؤ المهيكلة. الأعمال الحديثة وجدت أن أفضل تمثيل للكلمة يمكن الحصول عليه من خلال وصل أنواع مختلفة من التضمينات. إن عملية اختيار أنواع التضمينات التي ستدخل في عملية الوصل هذه متنوعة وتعتمد على المهمة التي نعمل عليها وما نتوقع أنه أفضل للمهمة. إن ظهور أنواع جديدة من التضمينات يجعل مهمة الاختيار هذه أصعب. يهدف هذا العمل لأتمتة عملية العثور على أفضل تسلسل من التضمينات لمهام التوقع المهيكلة من خلال نموذج يختار آليًّا التضمينات الأفضل. بمزيد من الدقة، يُستخدم نهج تكراري موجّه باستخدام مُتحكّم يأخذ في كل تكرار عينة من سلاسل التضمينات، وذلك اعتمادًا على اعتقاده الحالي بفاعليّة أنواع التضمينات التي شُكّلت منها السلسلة للمهمة التي يعمل عليها، ثم يُغذى نموذج المهمة بتمثيلات الكلمات هذه (تسلسل التضمينات الذي اختاره) ثم يُدرّب على بيانات المهمة، ويُرجع دقة النموذج كإشارة مكافأة يُحدّث من خلالها المتحكم اعتقاده بناءً عليها. تُستخدم آلية التحسين policy gradient algoritm [49] في التعليم المعزز لحل مشكلة التحسين (تحسين قرار المتحكم –اعتقاده–). من أجل تحسين عملية البحث تُستخدم أيضًا دالة مكافأة إضافية تعتمد على جمع كل المكافآت بناءًا على التحولات بين السلسلة الحالية وكل عينات السلاسل السابقة.

- **إيجابيات:**

حققت هذه الورقة نجاحًا كبيرًا في تحسين أداء العديد من المهام، حيث أُجريت عدة تجارب على 6 مهام و 21 مجموعة بيانات وأظهرت أن هذا النهج يتفوق في الأداء على خطوط أساس Baselines قوية، ويحقق أداءًا متطورًا SOTA عند استخدامه مع التضمينات المضبوطة بشكل خاص على المهمة (تضمين خاص بالمجال).

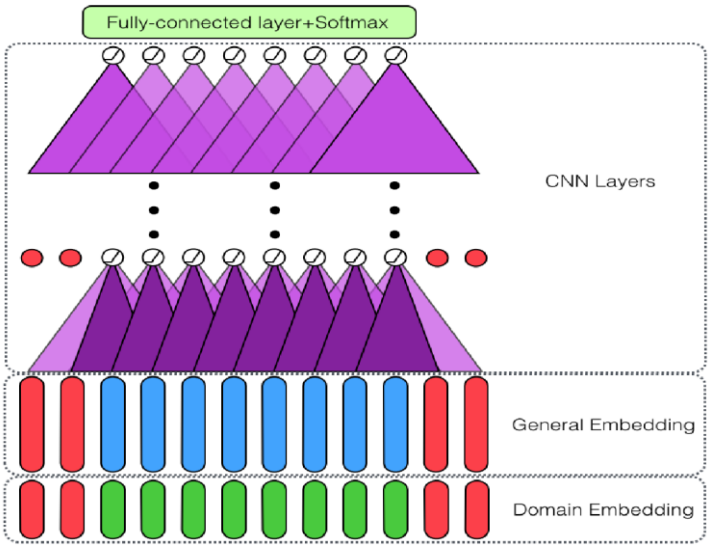
- **سلبيات:**

1. تمت صياغة ACE على أنها Neural architecture search problem، فهي تبحث عن أفضل وصل للتضمينات ضمن مساحة البحث المحددة. عندما نتحدث عن NAS فنحن نتحدث عن ساعات أو عشرات الساعات أو حتى آلاف الساعات من التدريب على GPU. تدعي الورقة أن النموذج يمكنه إيجاد تمثيل قوي للكلمات على GPU واحدة ببضع ساعات فقط وهذا جيد عندما نتحدث عن مشكلة تمت صياغتها ضمن NAS، لكن يبقى ذلك مكلفًا حوسبيًّا ويزداد الأمر سوءًا عند محاولة استخدامه مع البيانات الكبيرة جدًا من رتبة مئات الآلاف.

2. يمكن الاعتماد على خبرتك السابقة وفهمك للمهمة في اختيار أفضل أنواع التضمينات للمهمة المدروسة أو تجربة عدة تضمينات تُرشحها. مثلًا، من أجل مهمة NER يمكن أن نستبعد أنواع التضمينات المحرفية والاعتماد على التضمينات على مستوى الكلمات فقط لأننا لا نهتم كثيرًا بمورفيم الكلمات Morpheme هنا، بينما لو كانت المهمة هي ترجمة آلية أو Chatbots لاختلف الأمر.

❖ Double Embeddings and CNN-based Sequence Labeling for Aspect Extraction.
Authors: Hu Xu, Bing Liu, Lei Shu, Philip S. Yu.
Date: 11 May 2018

يقترح هذا العمل نموذج CNN بسيط، يستخدم نوعين من التضمينات المدربة مسبقًا لاستخراج الجوانب: تضمينات عامة وتضمينات خاصة بالمجال. النموذج المقترح الموضّح بالشكل (1–1) يحتوي على طبقتي تضمين و 4 طبقات CNN وطبقة FC مُشتركة عبر جميع مواضع الكلمات وطبقة Softmax تبعًا لفضاء الخرج لكل كلمة. يعتمد النموذج على ترميز IOB في ترميز الخرج (التسميات). طبقة التضمين الأولى تمثّل التضمين العام المُدرّب مسبقًا GLOVe على مجموعة نصية عامة الغرض (عادةً تكون مكونة من مئات الملايين من الكلمات). طبقة التضمين الثانية تمثل تضمينات المجال المدربة على مجموعة نصية صغيرة خاصة بالمجال FastText.



الشكل (1–1) نموذج DE–CNN

- **إيجابيات:**

1. حقق هذا النموذج أداءًا متطورًا في تلك الفترة، وكان أول نموذج يُقدّم تقريرًا عن نموذج CNN يعتمد على التضمينات المزدوجة لاستخراج الجانب.

2. يُظهر هذا النموذج أن استخدام تضمينات المجال إلى جانب التضمينات العامة يؤدي إلى تحسين الأداء.

3. على عكس الأعمال السابقة لمهمة AE التي اتسمت بالتعقيد، فإن هذا النموذج كان بسيطًا جدًا مقارنة بالأعمال السابقة وحقق نتائج أفضل.

4. النموذج خالٍ من أي إشراف إضافي ولا يستخدم معلومات إضافية تستند إلى المعاجم أو الميزات المصنوعة يدويًا.

- **سلبيات:**

1. النموذج يعتمد فقط على التضمينات الغير سياقية، وهذا ما سيحد من الأداء.

2. اعتماد نموذج CNN صافي فقط لمعالجة المهمة سيعطي محدودية في الأداء.

❖ Don't Eclipse Your Arts Due to Small Discrepancies: Boundary Repositioning with a Pointer Network for Aspect Extraction.
Authors: Zhenkai Wei, Yu Hong, Bowei Zou, Meng Cheng, Jianmin Yao.
Data: 5 July 2020

تتحدث الورقة عن مشكلة هامة في نماذج استخراج الجانب وهي الأخطاء الحدودية. تؤدي هذه الأخطاء إلى اختلاف بسيط نسبيًا بين الجوانب المستخرجة والجانب الصحيحة مما يؤدي إلى خفض الأداء. مثال:

Sentence: Their twist on pizza is healthy.

Ground-truth: twist on pizza

Predicted: pizza

تقترح الورقة استخدام "شبكة مؤشر" لإعادة ترتيب الحدود (إعادة تحديد بداية ونهاية الجانب). يمكن استخدام هذه الشبكة فوق أي شبكة استخراج جانب أخرى لمعالجة الأخطاء الحدودية فيها. تُدرّب هذه الشكبة بشكل منفصل على بيانات المهمة وتستخدم كمعالجة لاحقة لخرج نماذج استخراج الجانب.

- **إيجابيات:**

1. من خلال شبكة المؤشر يمكن تحسين الجوانب المحددة بشكل خاطئ.

2. تعمل شبكة المؤشر المدربة بشكل منفصل كمعالج لاحق، وبالتالي يمكن إقرانها بسهولة بنماذج استخراج جوانب مختلفة.

3. حققت هذه الورقة أداءًا متطورًا على العديد من مجموعات البيانات المعيارية.

- **سلبيات:**

1. يجب إعادة تدريبها من أجل مجموعات البيانات المختلفة.

2. لا تعطي دومًا نتائج أفضل.

3. تعتبر بمثابة إشراف إضافي.

❖ BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
Author: Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova.
Date: 21 June 2019

تُقدّم هذه الورقة أشهر نموذج لتمثيل اللغة حاليًا BERT (تمثيلات مُرمّز ثنائي الاتجاه باستخدام المحوّلات). يُقدم هذا النموذج اللغوي تمثيلات نص (تضمينات) سياقيّة تعكس المعنى الدلالي للكلمات بشكل غير مسبوق. دُرّب هذا النموذج على مهمتين مُختلفتين في نفس الوقت (50% لكل منهما) الأولى هي Masked Language Model والثانية هي Next Sentence Prediction باستخدام مجموعة نصيّة ضخمة من ويكيبيديا و BooksCorpus مكونة من 3.3 مليار كلمة. في المهمة الأولى يتم إخفاء 15% من الكلمات ضمن عينة نصيّة ما ويكون على النموذج مهمة توقع الكلمات التي أُخفيت وبالتالي يفرض MLM التعلم ثنائي الاتجاه من النص وهذا التكنيك لم يُستخدم من قبل! في المهمة الثانية يُستخدم NSP (توقع الجملة التالية) لمساعدة BERT على التعرف على العلاقات بين الجمل من خلال توقع ما إذا كانت جملة معينة تتبع الجملة السابقة أم لا. النموذج عبارة عن 24 كتلة Block من المحولات، كل منها يحتوي على 16 رأس انتباه Attention Head و1024 وحدة مخفية hidden state. عدد المعلمات التي يُدرّبها النموذج هو 340 مليون معلمة.

إيجابيات:

1. بسيط من الناحية المفاهيمية وقويًا من الناحية التجريبية.

2. تحليل الجمل وفهم السياق الذي كُتبت فيه.

3. يُقدّم تضمينات نص سياقيّة عالية الجودة تُحسّن الأداء بشكل كبير جدًا في المهام النهائية.

4. بيرت هو نموذج تعلم عميق يعتمد آلية الانتباه الذاتي المحولات Transformers؛ تتيح بنية المحولات إمكانية تفريع تدريب نماذج اللغة بكفاءة عالية. وبالتالي، فإن تفريع التدريب إلى حد كبير يجعل من الممكن تدريب BERT على كميات كبيرة من البيانات في فترة زمنية قصيرة نسبيًا (استخدام بُنى أخرى مثل LSTM مثلًا سيكون مُكلفًا حوسبيًّا مع هكذا بيانات).

5. قدم BERT حلًّا سحريًّا لأكثر من 11 مهمة من مهام اللغة الشائعة، مثل تحليل المشاعر وNER.

6. ساهمت مجموعات البيانات الكبيرة هذه في جعل BERT نموذج تضمين متفوق، ليس فقط في إنتاج تضمينات قوية، بل بفهم اللغة بحد ذاتها.

سلبيات:

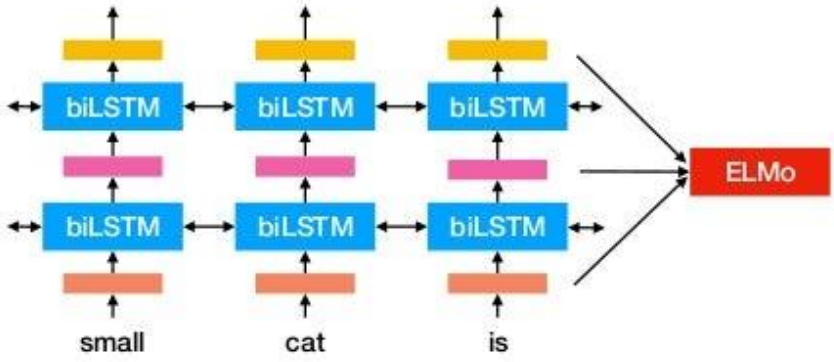
1. أُستُخدم لتدريب هذا النموذج 16 وحدة TPU (وحدات معالجة تفوق الواحدة منها سرعة 10 وحدات معالجة رسومية GPU). هذه الوحدات متوفرة فقط في جوجل (على جوجل كولاب –التابع لجوجل– يتم توفير وحدة TPU لكن بساعات محدودة)، وبالتالي من الصعب إعادة تدريبه من البداية أو ضبطه Fine–tuning من أجل مهمة محددة (للحصول على تضمينات المجال) لأننا في الغالب سنعتمد على وحدات GPU، وسيكون الأمر مكلفًا جدًا حوسبيًّا.

2. حجم النموذج كبير قليلًا (بحدود مئات الميغابايتات) مما قد يعيق استخدامه في البيئات الحوسبية الصغيرة (مثل الهواتف المحمولة وأجهزة الحاسوب العادية). لحل هذه المشكلة يتم اللجوء إلى نسخة أصغر حجمًا (على حساب الأداء) تتناسب أكثر مع هذه البيئات. كما تم تقديم نموذج آخر يُدعى DistilBERT كإصدار أخف منBERT؛ يعمل أسرع بنسبة 60٪ مع الحفاظ على أكثر من 95٪ من أداء BERT.

❖ Deep contextualized word representations.

Authors: Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer
Date: 24 June2018

تُقدّم هذه الورقة نموذج اللغة ELMo. تتبنى هذه الورقة طريقة جديدة لتضمين الكلمات. يتضمن الشكل (1-2) شبكة عصبية تلافيفية على مستوى الأحرف CNN لتمثيل كلمات السلسلة النصية ضمن متجهات أوليّة (تضمينات). تُغذّى طبقة BiLM بهذه المتجهات، يحتوي المسار الأمامي على معلومات متعلقة بكلمة محددة والسياق (كلمات أخرى) قبل تلك الكلمة. يحتوي المسار الخلفي على معلومات متعلقة بنفس الكلمة والسياق الذي يليها. يشكل هذا الزوج من المعلومات (من المسار الأمامي والخلفي) المتجهات الوسطية أو الثانوية. تُغذّى طبقة BiLM ثانية بهذه المتجهات الثانوية. التمثيل النهائي ELMo هو المجموع الموزون للمتجهات الأولية والمتجهات الثانوية. نموذج ELMO الأساسي يحتوي طبقتي BiLSTM كل طبقة تحتوي 4096 وحدة مخفية ويحتوي النموذج على93.6 مليون معلمة.



الشكل (1-2) نموذج ELMo

- **إيجابيات:**

1. بسيط من الناحية المفاهيمية وقويًا من الناحية التجريبية.

2. تحليل الجمل وفهم السياق الذي كُتبت فيه.

3. يُقدّم تضمينات نص سياقيّة عالية الجودة تُحسّن الأداء بشكل كبير جدًا في المهام النهائية.

4. حقق أداءًا متطورًا على ستة مشاكل صعبة عند استخدامه معها، منها الإجابة على الأسئلة والاستلزام النصي وتحليل المشاعر.

- **سلبيات:**

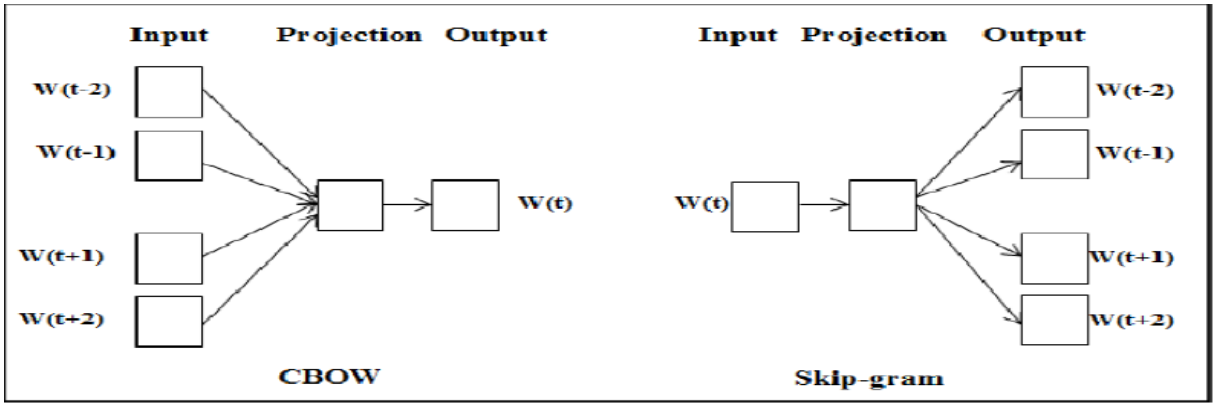
دُرّب باستخدام اثنين أو ثلاثة وحدات GPU من نوع Nvidia GeForce GTX 1080 Ti GPUs. واستغرق التدريب 3 أسابيع. هذا يقودنا إلى نفس المشكلة التي تحدثنا عنها في BERT، والمتعلقة بإعادة التدريب أو الضبط الدقيق على مجال محدد.

❖ Efficient Estimation of Word Representations in Vector Space

Authors: Tomas Mikolov & Kai Chen & Jeffrey Dean & Greg Corrado

Date: 7 SEP 2013

اقترحت هذه الورقة أول نموذج يُنتج تمثيلات نصية معتمدة على الشبكات العصبية تحت مفهوم "تضمينات الكلمات"، والذي يُشير إلى ربط كل كلمة بشعاع لتشكيل فضاء أشعة كل شعاع فيه هو إسقاط لكلمة ضمن هذا الفضاء، بحيث تكون العلاقات الهندسية بين هذه الأشعة انعكاس للعلاقات الدلالية بين هذه الكلمات. يأخذ هذا النموذج مجموعة كبيرة من النصوص كمدخلات ويتعلم تمثيل الكلمات في فضاء أشعة بناءًا على السياقات التي تظهر فيها واعتمادًا على أحد النهجين؛ الأول SkipGram والثانيCBOW في النهج الأول يُدرّب النموذج على توقع الكلمات المحيطة بالكلمة المحددة وفي النهج الثاني تكون المهمة توقع الكلمة المركزية بناءًا على الكلمات المحيطة، يوضّح الشكل التالي النموذجين:



الشكل (1-3) نموذج CBOW و SkipGram

- **إيجابيات:**

1. لقد كان لهذا النموذج آثار إيجابية هائلة في معالجة اللغة الطبيعية ولاسيما في التطبيقات التي تعتمد بشكل أكبر على المعاني الدلالية للكلمات، مثل تطبيقات الترجمة والسؤال والجواب والمحادثة الآلية وغيرهم، فقد كان قادراً على التقاط المعاني الدلالية للكلمات بشكل غير مسبوق إضافة إلى أنه كان تمثيلاً فعَالاً من الناحية الحوسبية فالأشعة كانت كثيفة Dense vector (بالكاد تحوي أصفار) وقليلة الأبعاد.

2. كان نقطة انطلاق لنماذج تضمين أفضل وأقوى ELMo وBERT وFlair وغيرهم. كما أنه أنها عصر استخدام تمثيلات النص السابقة مثل TF-IDF و N-grams وغيرها.

- **سلبيات:**

1. تضمينات الكلمات تُخزّن ضمن ملفات كبيرة الحجم (عدة غيغابايت)، والتي قد تسبب مشاكل في سيناريوهات نشر معينة. هذا شيء نحتاج إلى معالجته أثناء استخدامها، وإلا فقد يصبح عقبة هندسية في الأداء. يأخذ نموذج Word2vec حوالي 4.5 جيجابايت من ذاكرة الوصول العشوائي.

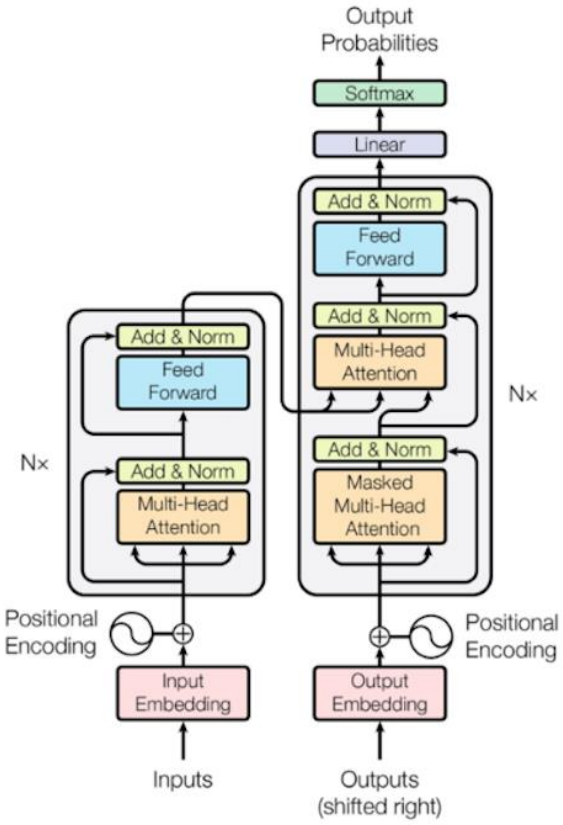
2. التحيّز؛ من المرجح أن يحدد نموذج التضمين الذي تم تدريبه بشكل كبير على أخبار أو مقالات التكنولوجيا أن Apple أقرب إلى Microsoft أو Facebook من البرتقال والموز.

3. نموذج تضمين غير سياقي. مثلًا، كلمة Python لها معاني مختلفة بناءًا على السياق الذي ظهرت فيه (أحيانًا تشير إلى "أفعى" وأحيان تشير إلى "لغة برمجة"). هذا النموذج لا يُراعي هكذا حالات، أي سيكون لكلمة Python نفس شعاع التضمين بغض النظر عن النص الذي تظهر فيه.

4. يعاني من مشكلة OOV أي أن الكلمات التي لم يُشاهدها ضمن مجموعة التدريب لن يتعرف عليها ولن يُعطي تمثيلًا لها.

❖ Attention Is All You Need.
Authors: Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser.
Date: 4 DEC 2017

تقترح هذه الورقة بنية شبكة جديدة تعتمد فقط على آلية الانتباه Attention Mechanism، مع الاستغناء التام عن العمليات التكرارية كما في RNN والتلافيفية كما في CNN. بنية الشبكة المقترحة تُسمى مُحوّل Transformer وهي موضحة بالشكل (1–4). وتهدف إلى تحقيق قفزة جديدة في مهام الترجمة الآلية.



الشكل (1–4) بنية المحوّلات

تتكون الشبكة من كتلتين الأولى (على اليمين) مُرمّز Encoder والثانية مُفكّك ترميز (على اليسار) Decoder، يُغذّى المُرمّز بتضمينات أولية للكلمات يُضاف إليها تضمينات موضعيّة (لفهم ترتيب الكلمات). تدخل هذه التضمينات إلى الكتلة الأولى والتي تتكون من عدة طبقات انتباه متعددة الرؤوس Multi–Head Attination مُكدّسة فوق بعضها (6 طبقات)، ثم إلى شبكة FC. الكتلة الثانية كتلة فك الترميز تأخذ نص

الخرج (الترجمة) وتطبق ذات العمليات التي تحدث في الكتلة الأولى ثم تدخل إلى 6 طبقات انتباه متعددة الرؤوس كما في الكتلة الأولى لكن مع إضافة فكرة الاخفاء Masking لبعض الكلمات. أيضًا هناك وصلات متسرّبة Risdual connecation من خرج الكتلة الأولى إلى دخل كل طبقة انتباه في الكتلة الثانية. تنتهي هذه الكتلة بطبقة FC تمثل الخرج.

- **إيجابيات:**

1. تُظهر التجارب التي أجريت على مهمتي ترجمة آلية أن هذه النماذج تتميز بجودة عالية.
2. بنية قابلة للتفريع بشكل غير مسبوق (كما تحدثنا في BERT) وتتطلب وقتًا أقل للتدريب.
3. أدت إلى قفزة كبيرة في الترجمة الآلية ومعالجة اللغات وانبثق عنها نماذج لغة مُدهشة مثل BERT و XLNet وغيرهم.
4. أُستُخدمت هذه البنية لاحقًا في العديد من المهام، حيث تم التعديل عليها بحيث تلائم المهمة المطلوبة.

- **سلبيات:**

العيب الرئيسي في طبقات آلية الانتباه (ليس المحوّل بشكل عام) هو أنها تضيف المزيد من الأوزان إلى النموذج، والتي يمكن أن تزيد من وقت التدريب خاصة إذا كانت بيانات الإدخال للنموذج عبارة عن تسلسلات طويلة.

❖ Enriching Word Vectors with Subword Information.
Authors: Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov
Date: 15 Jul 2016

يُقدّم هذا العمل نموذج اللغة FastText لتضمينات النص. هذا النموذج مشابه جدًا لنموذج Word2vec والذي تحدثت عنه. الفرق الأساسي هو أن هذا النموذج أتى بشكل أساسي ليحل مشكلة OOV التي ذكرتها سابقًا. لحل هذه المشكلة تستغل fastText معلومات الكلمات الفرعية لبناء التضمينات. يتم إنشاء تضمين الكلمة من خلال تجزيئها إلى n–grams (كلمات جزئية) وجمع تضمينات الكلمات الجزئية لتشكيل تمثيل الكلمة. هذا يساعد نموذج اللغة أيضًا على فهم اللواحق والبادئات ومورفيم الكلمات. fastText تم تدريبه على هذا الأساس؛ تمثيل الكلمة من خلال مجموع تضمينات أجزائها وتدريب نموذج skipgram لتعلم التضمينات.

- **إيجابيات:**

1. توسيع نموذج Word2vec السابق من خلال جعله يعمل على مستوى الكلمات الجزئية.
2. حل مشكلة OOV إلى حد كبير.

- **سلبيات:**

باستثناء OOV فهو يعاني من نفس مشاكل Word2vec.

1-3- البيئة والأدوات المستخدمة

البيئة المستخدمة هي بيئة بايثون من google colab المعتمدة على jupyter Nootebook. يسمح Colab لأي شخص بكتابة وتنفيذ كود Python من خلال المتصفح، وهو مناسب بشكل خاص للتعلم الآلي وتحليل البيانات والتعليم. ويقدم لنا موارد جيدة مجاناً لتساعدنا في تدريب نماذجنا، وتمنحنا تنفيذ الكود إما على ال CPU أو ال GPU أو على ال TPU.

لغة بايثون هي اللغة الأفضل والأهم عند التعامل مع مهام الذكاء الصنعي. نظراً لسهولة التعلم والاستخدام، يمكن كتابة أكواد بايثون وتنفيذها بسهولة أسرع بكثير من لغات البرمجة الأخرى وتتميز أيضا بالمكتبات الغنية بالأدوات التي لا غنى عنها ولاسيما في مجال تعلم الآلة، حيث أن أغلب أطر العمل يتم تشغيلها عليها مثل تنسرفلو وكيراس وباي تورش و Sklearn ومن أهم المكتبات المستخدمة في المشروع:

1. Tensorflow: هي إطار عمل مجاني ومفتوح المصدر في مجال تعلم الآلة. تستخدم في العديد من المجالات الفرعية ولكن لها تركيز محدد في تدريب واستدلال الشبكات العصبية العميقة.

2. Pytorch: هي إطار عمل مجاني ومفتوح المصدر في مجال تعلم الآلة. مستعمل لتطبيقات الرؤية الحاسوبية ومعالجة اللغات الطبيعية، مطورة أساسا من طرف مختبر آل أبحاث الفيسبوك.

3. Transformers: توفر هذه الحزمة واجهة برمجية لتنزيل وتدريب أحدث النماذج الجاهزة مثل BERT. يمكن أن يؤدي استخدام النماذج التي تم اختبارها مسبقًا إلى تقليل تكاليف الحوسبة وتوفير الوقت الذي تستغرقه في تدريب نموذج من البداية.

4. keras_nlp: هي حزمة برمجية تُقّدم مجموعة من الأدوات الإضافية مثل Tokinezers و Layers و Metrics إلخ، والتي تساعد الباحثين والمبرمجين على إنتاج نماذج NLP عالية الجودة.

5. tensorflow_addons: هي حزمة برمجية أخرى تُقّدم مجموعة من الأدوات الإضافية التي تساعد الباحثين والمبرمجين على إنتاج نماذج NLP عالية الجودة.

6. tensorflow_hub: هو مستودع مفتوح ومكتبة نماذج التعلم الآلي القابلة لإعادة الاستخدام (توفر نماذج مُدربة مسبقًا لاستخدامها مع المهام النهائية).

7. Gensim: هي حزمة مجانية مفتوحة المصدر لتمثيل المستندات كمتجهات دلالية (تضمينات)، حيث تضم تحقيقات عالية الكفاءة للعديد من نماذج التضمين الشهيرة، مثل Word2vec و FastText وغيرها.

8. Seqeval: هو إطار عمل لتقييم مهام وضع العلامات على التسلسل مثل NER وAE.

9. Sklearn: هي مكتبة برمجية مكتوبة بلغة البرمجة بايثون خاصة بمعالجة اللغات الطبيعية.

10. XML: حزمة برمجية تُسهّل التعامل مع الملفات المكتوبة بصيغة HTML وXML.

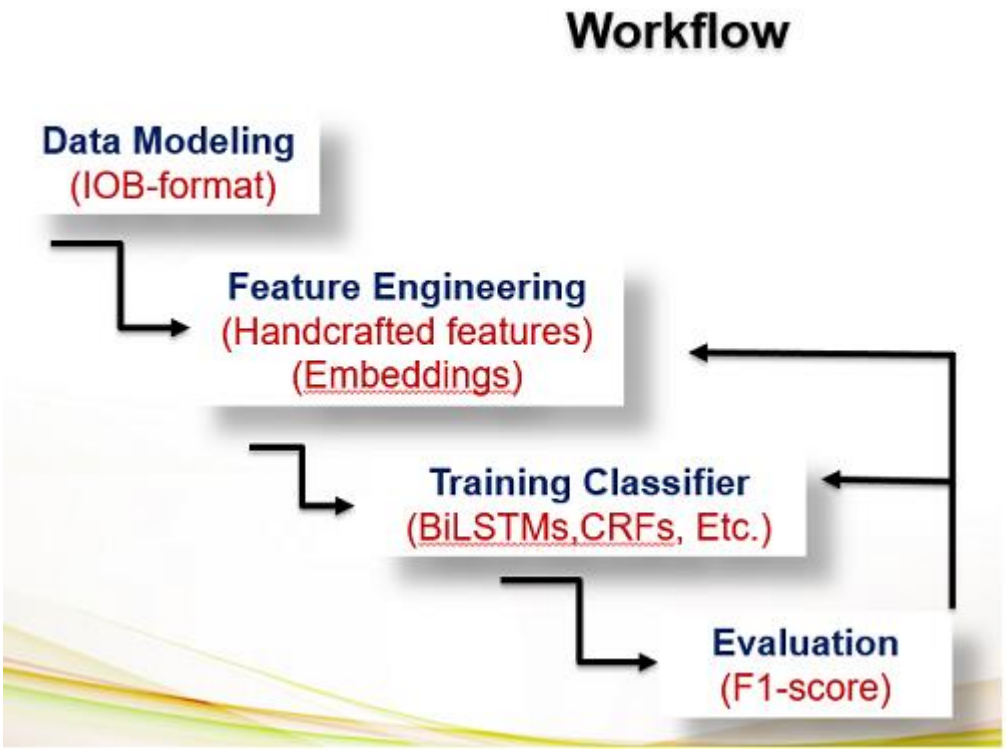
11. Numpy: تستخدم لتنفيذ عمليات الحوسبة العلمية. توفر إمكانية التعامل مع مصفوفات متعددة الأبعاد وبأداء عالي.

1-4- منهجية البحث

قُسّم هذا العمل إلى عدة مراحل:

1. المرحلة الأولى هي تحديد المشكلة وقراءة الأبحاث ذات الصلة بغية الإلمام بكامل خط تطور هذه المهمة.

   بعد الانتهاء من هذه المرحلة كان لدي فكرة واضحة عن الطرق التي قد تُحسّن أو تنافس أحدث ماوصلت إليه تلك الأعمال من خلال ملاحظة نقاط ضعفها وقوتها. وُثّق ذلك ضمن الفصل الأول، وعُرضت بإيجاز المواضيع والأعمال ذات الصلة في الفسل الثاني.
2. تضمّنت المرحلة الثانية تطبيق الأفكار المبنية من المرحلة الأولى، حيث اعتمدت على تطبيق نهج خاضع للإشراف وباستخدام شبكات CNN و LSTM و CRF و FC (يتم تجريب أكثر من طبولوجيا) يتم تغذيتها بتضمينات النص السياقية وغير السياقية الخاصة وغير الخاصة بالمجال (يتم تجريب عدة أنواع). اعتمدت على البروتوكول التالي في تطوير النموذج:



الشكل (1-5) بروتوكول تطوير النموذج

في الخطوة الأولى يتم تحويل البيانات إلى شكل يسمح بتدريب نموذج عليها، ثم في الخطوة الثانية يتم استخدام نماذج التضمين المُرشّحة لتمثيل النص ودمج هذه التمثيلات، ثم يُدرّب نموذج على هذه التضمينات ويُقيّم الأداء على بيانات الاختبار. يتم العودة إلى الخطوة الثانية والثالثة بشكل متكرر لإجراء تعديلات على النموذج وهندسة الميزات بغية تحسين الأداء. وُثّق كل ذلك بالتفصيل في الفصل الثالث.

3. تضمّنت المرحلة الثالثة توثيق أفضل نتيجة وصل إليها هذا العمل ومقارنته بنتائج الأعمال السابقة وذكر نقاط ضعفه وقوّته وكيفية تحسينه. وُثّق في الفصل الثالث.

2- استخراج الجوانب

تمت دراسة تحليل المشاعر على مستويات الجملة والمستند والجانب [29,30,3] تُركّز هذه الدراسة على مستوى الجانب. مثلًا، في مراجعات المنتجات، الجوانب هي سمات أو ميزات المنتج. مثلًا، لو كان المنتج هو لابتوب والجملة هي "Its speed is incredible" يكون الهدف هو استخراج "speed"، وتكون المشاعر المرتبطة بها إيجابية.

2-1- أعمال ذات صلة

يُعتبر استخراج الجانب واحدًا من أهم مهام هذه المهمة، وقد تمت تأدية هذه المهمة باستخدام كُلًّا من المناهج الخاضعة وغير الخاضعة للإشراف. يتضمن النهج غير الخاضع للإشراف طرقًا مثل Frequent Pattern Mining [31, 2] وهي عملية تحليلية تجد الأنماط المتكررة أو الارتباطات أو الهياكل السببية من مجموعات البيانات. syntactic rules-based extraction هي طريقة تعتمد على استخراج الجوانب بناءًا على قواعد نحوية مُحددة مسبقًا [32,4,6]. نمذجة الموضوع topic modeling وهي تقنية على مسح مجموعة من المستندات، واكتشاف أنماط الكلمات والعبارات داخلها، وتجميع مجموعات الكلمات والتعبيرات المماثلة التي تميز مجموعة من المستندات بشكل أفضل [33,34,5]. تقنية الكلمات المتحاذية word alignment [35]. يستخدم النهج الخاضع للإشراف [9,10,36] الحقول العشوائية الشرطيةConditional Random Fields [37]. في الآونة الأخيرة، يتم تطبيق الشبكات العصبية العميقة لتعلم ميزات أفضل، في المناهج الخاضعة للإشراف لمهام استخراج الجانب. مثلًا، استخدام LSTM [38,24] أو آليّة انتباه [15] جمبًا إلى جمب مع ميزات مُستخرجة يدويًّا [15,16]. كما أنه تم اقتراح استخراج الجوانب بالتوازي مع فئات الجوانب [15,13] من خلال الشبكات العصبية. تم استخدام شبكات CNN أيضًا، حيث لوحظ أنها تحقق نجاحًا في العديد من مهام معالجة اللغة الطبيعية أيضًا [25,26,27].

أحد العوائق الأساسية في LSTM هو أن خلاياها تعتمد على التسلسل، أي يجب أن يمر كل من الانتشار الأمامي Forward pass والانتشار الخلفي Backpropagation على كامل التسلسل، مما يؤدي إلى إبطاء

عملية التدريب والاختبار [1]. تتمثل إحدى العوائق التي تواجه CNN في مهام تصنيف (أو تسمية) السلاسل Sequence Labeling في أن عمليات الالتفاف Convolutional تؤدي إلى تقليص المدخلات التسلسلية والمخرجات لا تتحاذى جيدًا مع المدخلات [1]. الحقول العشوائية الشرطية هي فئة من أساليب النمذجة الإحصائية التي غالبًا ما تطبق في مهام التنبؤ المهيكلة. BiLSTM يمكنه فهم اللغة، بينما CRF يمكنه معرفة المنطق الضمني للتسميات. إذا كانت التسميات تتبع بنية داخلية محددة، فمن السهل جدًا أن يفهم CRF ذلك. في AE عادةً ما يتم ترميز المخرجات بثلاث رموز؛ بداية B وداخل I وخارجO ، حيث تشير B إلى بداية الجانب و Iإلى الأجزاء التالية منه و Oغير ذلك، ومن الواضح أنها تتبع بنية محددة. سيكتشف CRF بسرعة أنه من المستحيل على سبيل المثال أن يتبع O الرمز I، يجب أن يتبع B دائمًا. من ناحية أخرى قد يكون BiLSTM غير متأكد مما إذا كان يجب أن يضع B في موضع t أو t+1 وقد ينتهي به الأمر إلى إخراج كلاهما لأنهما مستقلان شرطيًا. طبقة CRF تعرف أن هذا غير محتمل وتفرض المنطق الداخلي للتسميات فتخرج B للأولى و I للتالية [43].

2-2- هندسة سمات البيانات النصية

بعبارة أخرى، كيف نحول نصاً معيناً إلى شكل رقمي بحيث يمكن إدخاله في خوارزميات معالجة اللغة الطبيعية وتعلم الألة. في معالجة اللغة الطبيعية، يسمى هذا التحويل للنص الخام إلى شكل رقمي مناسب تمثيل النص سنلقي نظرة على الطرق المختلفة لتمثيل النص، أو تمثيل النص كمتجه رقمي. فيما يتعلق بالصورة الأكبر لأي مشكلة في معالجة اللغة الطبيعية يمثل تمثيل السمة خطوة شائعة في أي مشروع ML، سواء كانت البيانات نصية أو صوراً أو مقاطع فيديو أو كلاماً. ومع ذلك، غالباً ما يكون تمثيل السمة للنص أكثر تعقيداً مقارنةً بتنسيقات البيانات الأخرى. هناك أسلوبين أساسيين يتم استخدامهم لتمثيل النص [40]:

- أساليب التمثيل الكلاسيكية (المعتمدة على العد).
- أساليب التمثيل المتقدمة (المعتمدة على التخمين).

2-2-1- أساليب التمثيل الكلاسيكية

هي أربعة طرق كانت تستخدم بشكل شائع لتمثيل النص، ولاسيما في الفترة التي سبقت ظهور نموذج Word2vec. طرق مثل One-Hot-Encoding وTF-IDF و Bag of Words و bag-of-n-grams أُستُخدمت لفترة طويلة لتمثيل النص، لكنها كانت تعاني من العديد من المشاكل فهمعظمها تمثيلات متناثرة Spare عالية الأبعاد وذريّة (تعامل الكلمات كوحدات مستقلة) وغير قادرة على التقاط المعنى الدلالي أو التشابه بين الكلمات هذا شكل عائقًا وتحديًا كبيرًا لمهام معالجة اللغة الطبيعية. أدى ذلك إلى ظهور ما يسمّى بنماذج التضمين [40].

2-2-2- أساليب التمثيل المتقدمة

إن الورقة البحثية التي نشرها ميكلوف و رفاقه أظهروا فيها أن نموذج تمثيل الكلمات المعتمد على الشبكة العصبية والمعروف باسم Word2vec، استناداً إلى "التشابه التوزيعي" [22]، يمكنه التقاط علاقات تشابه الكلمات مثل:

الملك – رجل + امرأة ≈ ملكة

كان نموذجهم قادراً على الإجابة بشكل صحيح على العديد من التشبيهات مثل هذا المثال. يعتبر نموذج Word2vec من نواحٍ عديدة فجر عصر معالجة اللغات الطبيعية الحديثة. ظهر بعد هذا النموذج العديد من التضمينات مثل FastText [39] وGlove [38] وجميعها كانت مشابهة إلى حد كبير بنموذج Word2vec. ساعدت هذه التضمينات مهام اللغة الطبيعية في إحراز تقدم كبير. هذه التضمينات اتصفت بأنها غير سياقيّة وتحتاج إلى ذاكرة كبيرة قليلًا للتخزين. ظهرت بعد ذلك التضميانات المحرفية [23] يتم تدريبها إلى جانب المهمة ويتم تطبيقها في العديد من مهام التنبؤ المهيكلة. ظهر بعد ذلك مفهوم التضمينات السياقيّة، وهي تضمينات تُمثّل الكلمة حسب سياقها التي ظهر فيه؛ كلمة Bank قد تعني في بعض الأحيان "ضفة نهر" وفي أحيان أخرى "بنك"، نماذج التضمين السياقية كانت قادرة على التقاط هذه الاختلافات على عكس نماذج التضمين السابقة. ELMo هو نموذج تضمين كلمات سياقي مُدرّب مسبقًا يعتمد على توليد التضمينات من خلال عدة طبقات BiLSTM ويتفوق بشكل كبير على أحدث الأساليب في العديد من مهام اللغة، هذا النموذج كان قادرًا على التقاط المعاني الدلالية للكلمات وتبعًا للسياق الذي تظهر فيه. ظهر بعد ذلك أيضًا نماذج تضمين مثل Flair [17] وهي نوع من التضمينات السياقية، وحققت أداءًا قويًّا في مهام وضع التسميات على التسلسل Sequence Labeling. ظهر بعد ذلك مفهوم جديد يُسمّى المُحوّلات [41] وهي بنى شبكات عصبية تعتمد على آليات انتباه متعددة الرؤوس، وحققت نجاحًا كبيرًا في مهام الترجمة الآليّة. كانت بنى المحولات الفكرة الأساسية وراء ظهور نماذج تضمين أحدث تعتمد على المحولات. BERT [19] هو نموذج تضمين سياقي يعتمد على المحولات وقد حقق نجاحًا كبيرًا جدًا. ظهر بعد ذلك نماذج مثل RoBERTa [42]، ركّزوا على تحسين نموذج BERT من خلال استراتيجيات إخفاء Masking أكثر قوة.

2-3- مخططات تسمية المخرجات (ترميز الجوانب)

تُعتبر مسألة استخراج الجانب مهمة مشابهة كثيرًا لمهمة التعرّف على الكيانات NER (كلاهما يندرج تحت اسم "تسمية التسلسل Sequence labeling"). إحدى النقاط التي يجب أن تؤخذ بعين الاعتبار في هكذا مهام هي طريقة ترميز المخرجات (سواءًا كانت جوانب في مهام AE أو كيانات في مهام NER). تعتبر مهمة NER مهمة أقدم وأكثر شهرة في معالجة اللغة الطبيعية، وظهرت العديد من مخططات التمثيل مثل IOB، IOB2، IOE1، IOE2، IO للتعامل مع هذه المهمة، وأظهرت الأبحاث أن مخطط التسمية IOB هو أفضل تمثيل ويعطي أفضل النتائج مع مهام NER. طرق التمثيل في AE أُخذت من NER كونها تُعتبر مهمة مشابهة جدًا، فاعتمدت معظم الأبحاث (التي أعرفها على الأقل) على مخطط التمثيل القياسي IOB كونه ملائم لهذه المهمة ولا يتضمن أي عيوب. تسبب ذلك في الابتعاد أو تجاهل استخدام مخطط التمثيل IO كونه يعتبر تمثيل يحتوي عيب بسيط.

يعتمد تمثيل IOB على ثلاثة رموز هي I داخل الجانب، O ليست جانب، B بداية الجانب. بينما يعتمد IO على رمزين فقط؛ I تشير إلى جانب، O ليست جانب، بالنسبة للجوانب المركبة (أكثر من كلمة) يتم اعتبارها سلسلة من الرمز I. تظهر المشكلة في تمثيل IO في أنه لن يكون قادرًا على الفصل بين الجوانب التي تكون متتالية (جانبين متتاليين مباشرةً)، حيث سيتم اعتبارهما جانبًا واحدًا. في الحالات العملية هذا نادر الحدوث (جانبين متتاليين)، ويمكن ملاحظة أنه يحدث في 99% من الحالات عندما يكون هناك خطأ قواعدي يمكن إصلاحه من خلال استخدام إشراف إضافي بسيط في مرحلة المعالجة المسبقة. [48]

3- تطوير نموذج SE-EM

3-1- مراحل العمل

3-1-1- المرحلة الأولى- التعريّف بمجموعة بيانات المهمة وتحليلها

يتم إجراء التجارب على ثلاث مجموعات بيانات تم اقتراحها في مسابقة SemEval عام 2014 و2016 وهي مسابقة سنوية مشهورة تتعلق بمهام معالجة اللغة الطبيعية. في مسابقة عام 2014 تم اقتراح مجموعتي بيانات الأولى مرتبطة بمجال المطاعم والثانية بأجهزة الحاسوب المحمولة SE-14 Task4 Subtask1. المهمة الأساسيّة هي تحليل مشاعر الجوانب، مُقسّمة إلى أربعة مهام جزئيّة. المهمة الجزئية الأولى تُمثّل موضوعنا في هذا البحث. مجموعة البيانات الثالثة هي من مسابقة عام 2016 وهي خاصة بمجال المطاعم SE-16 Task5 Subtask1. [44]

وصف المهمة: تُعطى مجموعة من الجُمل تُمثّل مُراجعات عملاء مرتبطة بالمطاعم أو أجهزة الحاسب. المهمة هي تحديد الجوانب الموجودة في كل جملة (مثلًا، waiter-appetizer- Cole slaw). يجب تحديد كل الجوانب في الجملة حتى ولو لم تكن هناك أي مشاعر مرتبطة بها، لأن ذلك سيكون مفيدًا لبناء أنطولوجيا مصطلحات الجانب ولتحديد الجوانب التي نوقشت بشكل متكرر. المهمة الجزئية الثانية يتم فيها إعطاء مصطلحات الجانب وتكون المهمة تحديد المشاعر المرتبطة بكل جانب (أمر معقد آخر لا يغطيه هذا العمل). المهمة الجزئية الثالثة والرابعة منفصلتين عن المهمتين الأولى والثانية وترتبطان باستخراج وتحليل مشاعر فئات الجوانب (فئات الجوانب أمر يختلف عن الجوانب) [44].

مجموعة بيانات المطاعم الأولى SE14-R تتكون من مجموعتين، واحدة للتدريب والأخرى للاختبار. تحتوي مجموعة التدريب على 3041 جملة إنجليزية مع الجوانب المرتبطة بها وتتضمن 3693 جانب، وتحتوي مجموعة الاختبار على 800 جملة إنجليزية مع الجوانب المرتبطة بها وتتضمن 1134 جانب. المجموعة الثانية SE16-R تحتوي على 2000 جملة وتتضمن 1744 جانب. تحوي مجموعة الاختبار على 676

جملة و622 جانب. مجموعة البيانات الثالثة SE14-L تحتوي على 3045 جملة للتدريب مع 2358 جانب، و800 جملة للاختبار مع 654 جانب. يتم تقييم أداء النماذج من خلال مقياس F1-Score [44].

the F_1 measure, defined as usually:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \quad (1)$$

where precision (P) and recall (R) are defined as:

$$P = \frac{|S \cap G|}{|S|}, R = \frac{|S \cap G|}{|G|} \quad (2)$$

قمت بإجراء بعض عمليات التحليل والاستكشاف للتعرف على هذه البيانات ورأيت أن الجوانب هي نوعين، فردية ومركبة. الفردية مكونة من كلمة واحدة وهي دومًا "اسم Noun" والمركبة مكونة من عدة كلمات تبدأ باسم وتنتهي باسم (parmesan cheese)، وقد يتوسطها أداة ربط أو حرف جر أو مُحدّدات واستخراجها سيكون أصعب بسبب وجود أحرف جر أو أدوات ربط (quality of the meat). لاحظت أيضًا أن بعض الكلمات يتم اعتبارها جانب في بعض الأوقات وفي أوقات أخرى لا واعتقد أن ذلك سيكون مشكلة وسيسبب حيرة للنموذج.

بما أنني اقترح تنسيق جديد لتسمية الجوانب IO والذي تم الحديث عنه في الفصل السابق، وجب أن أدرس حالة عدم التوافق. يمكن برهان أن التنسيق مُلائم تمامًا عدا حالة واحدة نادرة الحدوث وهي أن يكون هناك كلمتان متتاليتان تمثلان جانبان مختلفان وهذا لا يحدث إلا في حالة الأخطاء الإملائية بشكل عام. مثلًا، بالنسبة لمجموعة البيانات التي أتناولها والمؤلفة من 3841 جملة (تدريب+اختبار)، وجدت 26 حالة فقط يلتقي فيها جانبان متتاليان وجميعها أخطاء إملائية مُشابهة للمثال التالي، وقد تم توثيقها ضمن ملف data_analysis.py الذي يحتوي على تحليلات متعلقة بالتراكيب النحوية وبُنى الجوانب وحالات عدم التوافق. مثلًا في الجملة التالية:

The food drinks and service are clearly among the best in the city

في هذه الجملة يتم اعتبار كل من food و drinks على أنهما جانبان مختلفان، وبالتالي فإن ترميز IO لن يكون قادر على تفسير ذلك في هذه الحالة لأنه سيعتبر أن الكلمتان هما جانب واحد فقط، لكن وفقًا لقواعد اللغة كان يجب أن توضع فاصلة بعد كلمة food (خطأ إملائي) وبالتالي سيكون الترميز صحيحًا من خلال اعتبار الفاصلة Outside. أمر آخر يجب الانتباه له وهو أن ندرة حدوث هذه الحالة سيؤدي إلى تحيّز النموذج العكسي لهذه الحالة، أي بما أن 99.9 من الحالات لا يلتقي جانبان، فهذا يعني أن النموذج لن يكون قادرًا على تفسير هذه الحالة حتى لو تم تدريبه وفق ترميز IOB.

3-1-2-المرحلة الثانية- استخراج البيانات

تُقدّم مجموعات البيانات بتنسيق XML، وهي صيغة مُشابهة جدًا لصيغة HTML وفق الصيغة التالية:

```
<sentence id="11351725#582163#9">
  <text>Our waiter was friendly and it is a shame that he  didnt have a
supportive staff to work with.</text>
    <aspectTerms>
      <aspectTerm term="waiter" polarity="positive" from="4" to="10"/>
     <aspectTerm term="staff" polarity="negative" from="74" to="79"/>
    </aspectTerms>
    <aspectCategories>
       <aspectCategory category="service" polarity="conflict"/>
   </aspectCategories>
</sentence>
```

سنتحدث الآن عن مجموعة البيانات الأساسية التي أجري عليها تجاربي SE14-R ونفس الكلام ينطبق على باقي المجموعات.

بيانات التدريب مُخزّنة ضمن ملف Restaurants_Train_v2.xml، وبيانات الاختبار ضمن ملف Restaurants_Test_Gold.xml. من هذين الملفين يتم إنتاج ملفان جديدان هما Restaurants_Train_v2_mod.iob و Restaurants_Test_Gold_mod.iob يُقابلان ملف التدريب والاختبار على التوالي. يتم إنجاز ذلك من خلال ملف adapter.py. اعتمدت على الحزمة xml التي تحتوي صنف Class يُدعى ElementTree يُمكّنك من قراءة وتحليل ملفات XML. أقوم باستخدام هذا الصنف واستخرج البيانات من هذه الملفات ثم اكتبها ضمن ملفات جديدة بتنسيق IOB. مثلًا، أقوم بكتابة الجملة التالية ضمن ملف الخرج وفق الصيغة التالية، وأفصل الجمل عن بعضها بسطر فارغ:

Sentence: But the staff was so horrible to us

Output:

But O

the O

Staff B-A

was O

so O

horrible O

to O

us O

Our O

agreed O

..

..

Orrechiete B-A

with I-A

sausage I-A

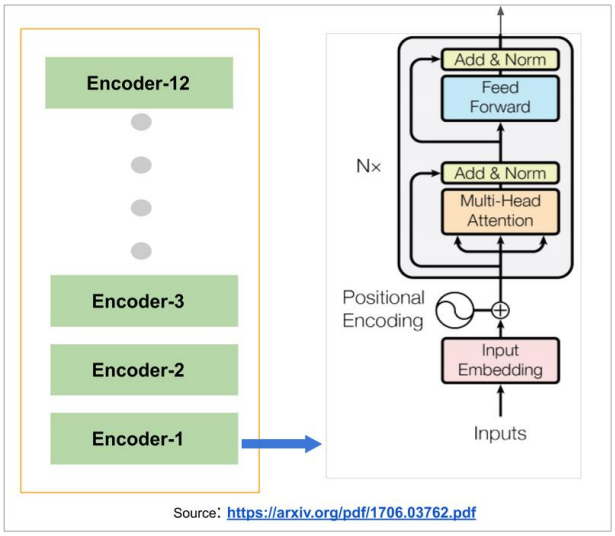
and I-A

chicken I-A

3-1-3- المرحلة الثالثة- استخراج السمات من نموذج اللغة BERT المُدرّب مُسبقًا

قمت سابقًا بتوصيف معمارية هذا النموذج، حيث ذكرت أنه يتكون من قسمين مُرمّز Encoder ومفكك ترميز Decoder وأنه مُدرّب على مليارات الكلمات. ذكرت أيضًا أنه بإمكاننا استخدامه لتمثيل (تضمين) كلمات المهمة، لكن كيف ذلك؟

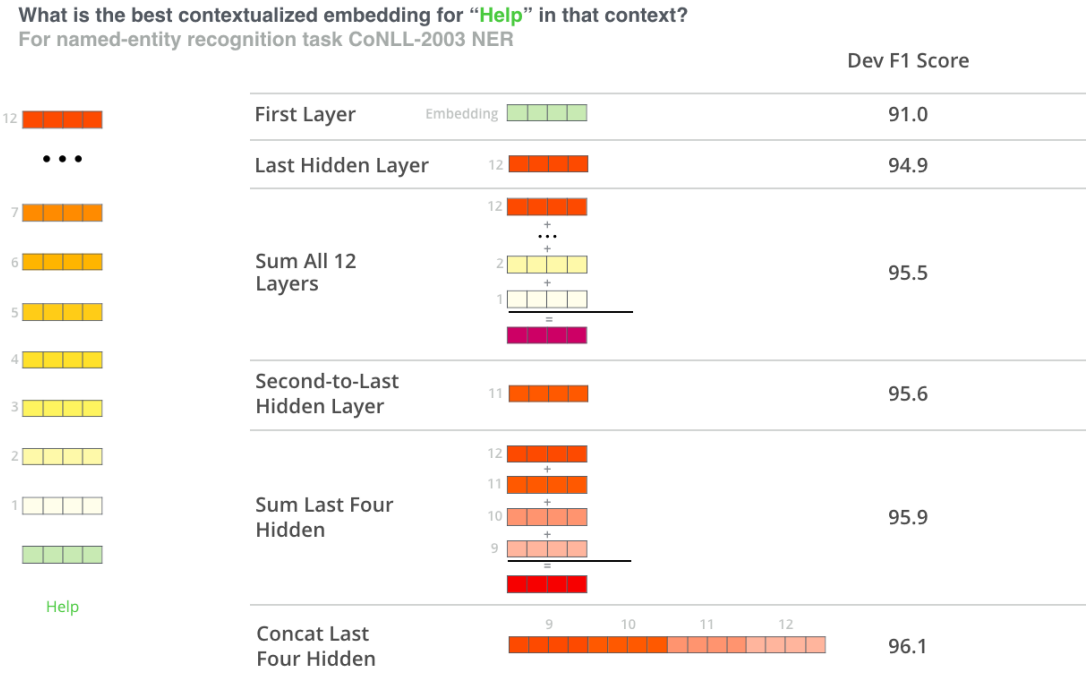
علينا أولاً فهم معمارية وآلية عمل النموذج جيدًا من خلال قراءة الورقة [19]. هناك نسختان من هذا النموذج الأولى هي larg والثانية base، سأتعامل هنا مع النسخة الثانية لأنها أخف. لاستخراج تمثيلات الكلمات سنتعامل مع جزء Encoder فقط، والموضّح بالشكل (3-1):



الشكل (3-1) المُرمّز Encoder

المُرمّز يتألف من 12 مُرمّز، كل منها يحوي طبقة انتباه مُتعددة الرؤوس تحتوي على 786 وحدة مخفية تُمثّل أبعاد الخرج (هذه الوحدات ستمثّل التضمينات). يُمكن استخدام أي طبقة من هذه الطبقات للحصول على تمثيلات الكلمات الموجودة في بياناتنا. إن عملية تقرير أي من هذه الطبقات ال 12 سيكون تمثيلًا لبياناتنا هو موضوع أمثلية بحد ذاته. وجد الباحثين من خلال التجربة أن أفضل النتائج تكون من خلال جمع مُخرجات آخر 4 طبقات أو وصل خرج آخر 4 طبقات (بالتالي نحصل على تمثيل كلمة أبعاده 4*786).

الطريقة الأولى هي التي اعتمدتها لأن الطريقة الثانية تزيد الكلفة الحوسبية 4 أضعاف. تجدر الملاحظة إلى أن الطريقة الثانية أثبتت أنها تُحسن النتائج بفارق أجزاء من المئة من خلال التجارب (فارق لايُذكر مقارنةً بالكفلة الحوسبية التي ستنتج منها)، الشكل (3–2) التالي يوضح المقارنة:



الشكل (3–2) مقارنة بين طرق الدمج المختلفة لطبقات التضمين في نموذج BERT

يتضمن ملف bert_features.py شيفرة برمجية قمت ببنائها لاستخراج تمثيلات الكلمات وفق الآلية التالية، كما يتضمن الشيفرة اللازمة لعمل Fine-tuning لنفس النموذج على مجموعة بيانات خاصة بالمجال:

1. نستخدم الحزمة transformers لتحميل نموذج بيرت المُدرّب مُسبقًا.
2. أقوم بتعريف الصف BERTModelFeatures وأُهيّئه بالمعلومات الأولية المطلوبة وهي النموذج المُدرّب و WordPiece (نموذج تقسيم نصوص إلى وحدات Tokenaization) [45].
3. بما أن نموذج بيرت هو نموذج مُدرّب مسبقًا، فهو يتوقع مُدخلات بصيغة محددة:
   a. محرف خاص [SEP] للفصل بين الجمل.
   b. محرف خاص [CLS] في بداية النص.

[CLS] The man went to the store. [SEP] He bought a gallon of milk.

c. الوحدات Tokens وهي النص المُقسّم إلى كلمات وفقًا لنموذج WordPiece (إذا كانت الكلمة غير موجودة يتم تقسيمها إلى كلمات جزئية وفقًا لآلية خاصة). مثلًا:

Sentence: text = "Here is the sentence I want embeddings for."
Tokenization:
['[CLS]', 'here', 'is', 'the', 'sentence', 'i', 'want', 'em', '##bed', '##ding', '##s', 'for', '.', '[SEP]']

لاحظ أن كلمة embeddings تم تقسيمها وفق التالي:

['em', '##bed', '##ding', '##s']

إنها آلية بيرت لمعالجة مشكلة OOV. نموذج WordPiece لم يتعرّف على الكلمة ضمن قاموس مُفرداته (قاموس الكلمات التي يعرفها بيرت)، فقام بتقسيمها إلى كلمات جُزئية يعرفها. الرمز ## يُشير إلى أن هذه

الوحدة تنتمي إلى الوحدة السابقة. لاحقًا، يجب دمج تضمينات هذه الكلمات لتشكيل تمثيل كلمة embeddings.

d. مُعرّفات الوحدات Token Ids، تُربط كل وحدة بمُعرّف (عدد صحيح فريد يولده WordPiece).
e. مُعرّفات الإخفاء Mask ids ، شعاع يربط كل وحدة بقيمة 1 أو 0، في حال كانت 0 فهذا يعني أنها غير مهمة وعلى النموذج تجاهلها (عادةً تكون كلمات الحشو Padding).
f. مُعرّفات الأجزاء Segment Ids، بيرت تم تدريبه على مهمتين مختلفتين بنفس الوقت كما أشرنا في الفصل الأول. هذا الشعاع يستخدم للتفريق بين الجملة الأولى والثانية. الجملة الأولى تكون مُعرفاتها 1، والجملة الثانية تكون معرفاتها أصفار.
g. تضمينات موضعية Positional Embeddings. آلية الانتباه لا يمكنها فهم التسلسل (ترتيب الكلمات في النص)، لذا يتم إضافة فكرة التضمينات الموضعية التي تُجمع مع التضمينات الأولية قبل أن تُمرر لأول طبقة انتباه. هذه التضمينات تمنح آلية الانتباه القدرة على التقاط السياق (التسلسل).

4. نقوم بعد ذلك بتمرير النص وفقًا للصيغة السابقة إلى نموذج بيرت، والذي يقوم بدوره بتمرير هذا النص عبر الطبقات ال 12.
5. بالنسبة لكل جملة (عينة) لدينا مايلي:
   a. 13 طبقة (12+طبقة التضمينات الأولية).
   b. حجم الحزمة 1 (عدد الجُمل) التي نمررها للنموذج (نتحدث عن جملة واحدة).
   c. عدد الوحدات في كل جملة. بما أن الجمل تحوي أعدادًا مُختلفة من الكلمات، نقوم بحشو الجمل التي تحوي أقل من 70 كلمة، حيث 70 هو عدد كلمات أطول جملة في مجموعة بيانات التدريب والاختبار.
   d. عدد الوحدات المخفية (الميزات) في كل طبقة انتباه 768.

e. خرج كل طبقة سيكون:

[# layers, # batches, # tokens, # features]

الخرج الذي نريده:

[# tokens, # layers, # features]

ننجز عملية التحويل هذه من خلال الدالة permute في حزمة بايتورش.

6. بعد الحصول على خرج كل جملة نقوم بجمع مُخرجات آخر 4 طبقات من أجل كل وحدة لنحصل على الخرج النهائي لكل كلمة ضمن هذه الجملة.

ملاحظات:

1. هذا نموذج سياقي؛ أي أنه يعطي الكلمة تمثيلًا وفقًا للسياق الذي تظهر فيه، وبالتالي كلمة bank إذا ظهرت في سياق يتحدث عن المال أو شيء من هذا القبيل، فسيكون التمثيل مُختلفًا فيما لو ظهرت في سياق يتحدث عن الطبيعة أو النهر (يكون القصد منها "ضفّة"). بنيت دوال إضافية –يمكن أن يستفاد منها الآخرين– يمكن استخدامها لحساب التشابه بين الجمل أو الكلمات وفق مقياس تشابه جيب التمام Cosine Similarity، كما بنيت دالة أخرى تمكنك من وصل تضمينات آخر 4 طبقات بدلًا من جمعها. مثلًا، لو كان لدينا النص التالي:

“After stealing money from the bank vault, the bank robber was seen fishing on the Mississippi river bank.”

لو مررنا هذه الجملة لنموذج بيرت وحصلنا على تمثيلات الكلمات منها، ثم حسبنا التشابه بين تمثيلات كلمة bank وفق مقياس تشابه جيب التمام (يمكن حساب كل هذه الأمور من خلال نموذج BERTModelFeatures الذي بنيته) سيكون الخرج كما يلي:

Vector similarity for *different* meanings: 0.69
Vector similarity for *similar* meanings: 0.94

الخرج الأول يمثل تشابه الأشعة التي تمثل كلمة bank في السياقين "river bank" vs "bank robber"".
والثاني في السياقين "bank vault" vs "bank robber". هذا يُظهر مدى روعة بيرت وقدرته على فهم اللغة!

2. الكلاس BERTModelFeatures بنيته بطريقة يمكنك التعديل عليه بسهولة تامة للحصول على أي شيء تريده (جمع أكثر من 4 طبقات، وصل الطبقات، الحصول على تمثل كلمة واحدة فقط ضمن سياق محدد، حساب تشابه الكلمات، تشابه الجمل ..إلخ).
3. كما ذكرت سابقًا، الكلمات خارج القاموس يتم تقسيمها إلى كلمات جزئية يعرفها النموذج، وبالتالي يجب إعادة تشكيل تضمين الكلمة الأساسية من تضمينات الكلمات الجزئية يدويًّا، وهذا يتطلب معرفة كل كلمة قام النموذج بتقسيمها ثم معرفة تضمينات الكلمات الجزئية ومواضعها ثم حذف تضمينات الكلمات الجزئية بعد دمجها في تضمين واحد (هذا ضروري جدًا). لقد قمت ببناء شيفرة تحل هذه المشكلة بشكل دقيق فلا داعي للقلق، حيث اعتمدت على جمع هذه التضمينات. يمكنك استخدام تكنيكات أخرى؛ مثل اعتبار أول كلمة جزئية كتضمين للكلمة الأساسية وحذف البقية أو تضمين الكلمة الجزئية الأخيرة، لكن اعتقد أن الجمع بين الكلمات أفضل لأنه لا يفقد المعلومات.
4. التضمينات التي نحصل عليها لكل كلمة من خلال هذا النموذج أبعادها 768 بالنسبة للتضمينات العامة و 1024 بالنسبة للتضمينات الخاصة بالمجال.
5. يتم عمل Fine-tuning لنفس النموذج على بيانات خاصة بالمجال لاستخدامه.

3-1-4- المرحلة الرابعة- استخراج الميزات من نموذج اللغة ELMo المُدرّب مُسبقًا

بنيت الصنف elmo_features.py لاستخراج تمثيلات الكلمات من نموذج إلمو. استخدم الحزمة tensorflow_hub للحصول على النموذج وتحميله. النموذج يتعامل مع الكلمات على مستوى المحارف، وبالتالي يحل مشكلة OOV تمامًا، فلا داع للقلق من الأمر. يمكنك تمرير النص كما هو وستحصل على المخرجات مباشرةً من خلال الصنف ELMoModelFeatures (يتضمن القيام بالعمليات الروتينية مثل الحشو والتقسيم إلى وحدات).

ملاحظة: التضمينات التي نحصل عليها لكل كلمة من هذا النموذج أبعادها 1024.

3-1-5- المرحلة الخامسة- استخراج السمات من نموذج اللغة FastText المُدرّب مُسبقًا

الملفين السابقين استخدمهما لتوليد تضمينات النص العامة السياقية. أقوم باستخدام نموذج FastText مُدرّب مُسبقًا على بيانات مُرتبطة بالمطاعم، هذا النموذج قام بتدريبه Xu et al., 2018 لاستخدامه في نموذجهم [1]. النماذج غير السياقية ينتج عنها قاموس مُفردات بكل الكلمات التي رأتها في المجموعة النصية التي دُرّبت عليها مع التضمين المرتبط بها. هذا التضمين ثابت وتحصل عليه من القاموس النهائي وهو نفسه بغض النظر عن السياق الذي تظهر به الكلمة. أي كلمة bank سيكون لها نفس الشعاع دومًا (على عكس بيرت وإلمو التي تراعي السياق).
كما هو معروف فإن هذ النموذج يعاني من مشكلة OOV أيضًا (يعالجها لكن ليس تمامًا). لذا عالجت هذا الأمر من خلال تدريب نموذج FastText آخر من البداية على مجموعة بيانات مُشابهة لمجموعة بيانات المطاعم التي أعمل عليها تتألف من مليون مُراجعة وتحتوي مجموعات البيانات التي أعمل عليها (هذا لايتضمن مجموعات الاختبار). أقوم بتدريب النموذج بنفس الطريقة التي دُرّب عليها النموذج المُدرب الذي استخدمه.

يحتوي الملف fast_text_train_model.py على التعليمات البرمجية اللازمة لتدريب النموذج. استخدم الحزمة genism التي تتضمن تحقيق عالي الكفاءة لهذا النموذج، ومجموعة بيانات amazon-fine-food-

reviews لتدريب النموذج جمبًا إلى جمب مع بيانات المهمة نفسها. أُخزّن النموذج الناتج fast-text-model لاستخدامه في إنتاج التضمينات.

في الملف prep_indomain_emb.py أقوم بقراءة البيانات من ملفات iob التي أنتجناها سابقًا، ثم نولّد ملف جسون worldx_ file.json يُمثّل قاموسًا بكل الكلمات الموجودة في بيانات المهمة. بعد ذلك أقوم بربط كل كلمة في القاموس بعدد صحيح يُمثّل فهرس سنستخدمه لاستخلاص تضمين هذه الكلمة من مصفوفة التضمينات التي سأنشئها والتي سنحصل عليها من النموذج المُدرّب.

بعد ذلك نقوم بتعريف الدالة gen_np_embedding التي تنشئ مصفوفة التضمينات وتكتبها ضمن ملف restaurant_emb.npy. تُنشئ هذه الدالة أيضًا ملف restaurant_emb.oov.txt يحتوي على كل الكلمات التي لم نجد لها تضمينًا في النموذج المُدرّب مسبقًا.

الآن من خلال الملف oov_prep.py استخدم ملف restaurant_emb.npy وملف الكلمات خارج القاموس restaurant_emb.oov.txt والنموذج الذي درّبته fast-text-model لإنشاء تضمينات لتلك الكلمات. ينتج من هذا الملف التضمينات النهائية الخاصة بالمجال التي نحصل عليها من FastText، حيث تُخزّن ضمن ملف final_indomain_emb.npy.

ملاحظات:

1. يمكنك استخدام أي طريقة أخرى لمعالجة مشكلة الكلمات خارج القاموس وليس الاعتماد على هذه الطريقة، لكن وجدتها أنسب، فأنا أُدرب نفس النموذج على بيانات مُشتقة من البيانات التي دُرّب عليها النموذج الأصلي وأضيف مجموعة بيانات المهمة لها بحيث تكون جميع الكلمات معروفة.
2. قد يتبادر للأذهان الاستفهام التالي، لماذا لم أقوم بتدريب نموذج إلمو للحصول على تضمينات المجال بدلًا من الاعتماد على نموذج أبسط مثل FastText. فكما نعلم، تضمينات إلمو أكثر فعالية. الإجابة على ذلك هو الكلفة الحوسبية، فإعادة تدريب إحدى هذه النماذج من البداية أو ضبطه لمهمة المجال سيستهلك عشرات الساعات المتواصلة على GPU سريعة. حتى أنه لا يمكنك الاعتماد على الخادم سيزر الذي يُقّره قسم الذكاء الصنعي نظرًا لمحدوديته (8 غيغابايت سعة الذاكرة) لتدريب هكذا نماذج (ستحتاج أيام عديدة). لذا فتدريب FastText أسرع بكثير.

3-1-6- المرحلة السادسة- دمج التضمينات السياقية وغير السياقية

في الملف stacked_features&prep_data.py نقوم باستخدام الكلاس BERTModelFeatures من الملف bert__features.py والكلاس ELMoModelFeatures من الملف elmo_features.py والملف final_indomain_emb.npy لوصل التضمينات وإنتاج التمثيلات النهائية. نقوم بقراءة بيانات التدريب والاختبار ونقوم بتهيئة الكلاسات السابقة وتمرير المعطيات إلى الدالة stacked_emb التي تقوم بوصل هذه التمثيلات وإنتاج بيانات التدريب النهائية. نقوم أيضًا بإنتاج بيانات الخرج test_labels.npy و training_labels.npy.

ملاحظة: تضمينات إلمو تعطي تضمينًا أبعاده 1024، بيرت 768، FastText يعطي 100، وبيرت المُعاد ضبطه 1024. عند تكديس هذه التضمينات نحصل على تضمين كلمة طوله 2916=768+100+1024+1024.

3-1-7- المرحلة السابعة- إنشاء ملفات تدريب النموذج

أقوم بتدريب نموذجين مُتطابقين باستثناء طريقة ترميز الجوانب؛ في النموذج الأول استخدم تنسيق IOB الكلاسيكي، وفي النموذج الثاني استخدم تنسيق IO. الهدف من ذلك أمرين؛ إثبات فعاليّة النموذج حتى باستخدام التنسيق الكلاسيكي، والثاني أفضلية استخدام تنسيق IO وحتى بدون استخدام إشراف إضافي. بدايةً هناك عدة ملفات أنشأتها يجب الحديث عنها.

1. ملف callbacks.py هو ملف يحتوي كلاس اسمه Monitor يُمثّل رد اتصال "callback" مُعرّف يدويًا. أهدف من خلاله إلى إظهار معلومات مُتعلقة بتدريب النموذج والقيام بإجراءات أخرى، وهي:
   a. حساب عدد المرات التي تنبأ فيها النموذج ببداية الجانب بشكل صحيح وحساب دقة النموذج في ذلك.

b. حساب عدد المرات التي تنبأ فيها النموذج بالجانب (كاملًا) بشكل صحيح وحساب دقة النموذج في ذلك.
c. حساب درجة F1-Score وحساب Recall-Score وحساب Precision-Score.
d. التوقف المُبكّر Early stopping؛ عندما يتوقف النموذج عن التحسن خلال عدد محدد من epochs، يتم إيقاف التدريب.
e. حفظ النموذج، وهنا وضعت خيارين؛ الأول حفظ نسخة من النموذج بعد كل epoch (نقاط اختبارية Checkpoints) والثاني حفظ أفضل نموذج حصلنا عليه خلال التدريب فقط (يتم الحفظ على أساس درجة F1-Score).

ملاحظات:

1. هذا الملف يُجري حساباته على أساس بيانات تتبع فعاليّة النموذج Validation set (يُشاع تسميتها "بيانات التحقق من الصحة"، لكن أراها ترجمة حرفية).

2. تتبع دالة الخسارة غير مُجدي فالنموذج يحتوي على بيانات غير متوازنة بشكل كبير؛ عدد الكلمات التي لا تمثل جانب، أكبر بكثر من الكلمات التي تُمثّل جانب، وبالتالي سيكون هناك تحيّز كبير جدًا في نتيجة دالة الخسارة (أعالج هذا الأمر إلى حد كبير من خلال استخدام دالة خسارة خاصة). كما أن الاعتماد على نتيجة الدقة هو أمر غير دقيق أيضًا لنفس السبب، ولأنها لا تراعي أخطاء النموذج في التوقع.

3. يجب الاعتماد على درجة F1-Score وتتبعها، فهي المقياس الأدق.

4. لابد من تعرف الكلاس Monitor، لأن إطار العمل تنسرفلو أو حتى باي تورش لا يوجد فيهما تحقيقات جاهزة لهذه المهام.

5. يتم استدعاء Monitor بعد نهاية كل epoch، لتقييم النموذج وفق المعايير السابقة.

2. ملف layer.py، يحتوي هذا الملف على تحقيق لطبقة تمثل آلية انتباه، حيث استخدمت هذه الطبقة خلال تجاربي في الحصول على أفضل نموذج.
3. ملف Metrics.py، يحتوي على تحقيقات لحساب كل من الدقة ودرجة F1-Score.. إلخ. يتم تعريف هذا الملف من أجل تقييم النموذج النهائي على بيانات الاختبار.
4. ملف data_analysis.ipynb، ذكرته في بداية الفصل، يحتوي على تحليلات أجريتها على البيانات.
5. ملف data_generator.py، يحتوي هذا الملف على الكلاس DataGenerator. الهدف من هذا الكلاس هو استخدام الذاكرة بشكل أمثلي. بدلًا من تخزين كامل مجموعة البيانات في الذاكرة ثم تقسيمها لحزم وتدريب النموذج عليها (تحتاج إلى ذاكرة GPU حجمها أكبر من 12 غيغا في هكذا حالة) يتم تركها على القرص ويتم تحميّل الحزمة المطلوب فقط في كل تكرار ضمن كل epoch.

بمزيد من التوضيح؛ لدينا 2890 عينة تدريبية لو حددنا حجم كل حزمة ب 32 (أي 32 عينة)، سيكون لدينا 90 حزمة (32/2890). كل epoch عبارة عن مرور على كامل البيانات أي المرور على 90 حزمة. المرور على حزمة واحدة نسميه تكرار، وفي كل تكرار يتم إجراء عملية انتشار أمامي وانتشار خلفي (تحديث الأوزان وفق خوارزمية انحدار المُشتق GD). بعد انتهاء ال epoch (أي المرور على كامل البيانات -90 حزمة-) نُعيد نفس العملية، وهكذا حتى انتهاء عدد ال epochs المحدد أو حدوث توقف مُبكر. نلاحظ أنه بدلاً من الإبقاء على كامل الحزم في الذاكرة يمكنك تركها على القرص وترك الحزمة المطلوبة في الخطوة t فقط. بعد إنجاز عملية الانتشار الامامي والخلفي على الحزمة 1b، نُحمّل الحزمة 2b وهكذا. أي في كل خطوة زمنية يكون لدينا حزمة واحدة فقط في الذاكرة بدلًا من إبقاء كامل الحزم واستهلاك ذاكرة لا داع لها.

ملاحظة: يتطلب استخدام هذا التكنيك حفظ البيانات وفق صيغة محددة. ننشئ قاموس ندعوه مثلاً partition.

- in partition['train'] a list of training IDs
- in partition['validation'] a list of validation IDs

  {'train': ['id-1', 'id-2', 'id-3'], 'validation': ['id-4']}
- Create a dictionary called labels where for each ID of the dataset, the associated label is given by labels[ID]

  {'id-1': 0, 'id-2': 1, 'id-3': 2, 'id-4': 1}

3-1-8- المرحلة الثامنة- النموذج

يتم تجربة نفس النموذج الشكل (3-3)، لكن مع تغيير ترميز الجوانب، مرة نجربه مع تمثيل IO ومرة مع IOB وأقوم بتجربته أيضًا على 3 مجموعات بيانات مختلفة. يتألف النموذج من كتلتين منفصلتين؛ الأولى مُكونة من 3 طباقات BiLSTM مُكدسة فوق بعضها، والثانية مكونة من تكديس 11 طبقة CNN. كُل من هاتين الطبقتين تتلقى نفس المُدخلات التي تمثّل التضمينات المُكدّسة والتي سبق وتحدثنا عنها. يُدمج خرج الكتلتين السابقتين خطيًا، وتُغذّى به طبقتي اتصال كامل FC بأوزان مُشتركة عبر كل مواضع الكلمات. أخيرًا يتم تغذية طبقة الخرج والممثلة بطبقة CRF بمخرجات طبقة الاتصال الكامل.

بالنسبة لتمثيلات IOB يمكن أن يكون خرج كل كلمة إما B أو I أو O، أما بالنسبة لتمثيل IO فهو إما I أو O. لنفترض أن دخل الشبكة هو سلسلة من الكلمات:

x=(x1,x2,..,xn)

تحصل كل كلمة من هذا التسلسل على أربعة تضمينات xf xe xb1 xb2، التضمين الأول يمثّل التضمين الخاص بالمجال والثاني والثالث التضمينات العامة والرابع تضمين خاص بالمجال كما ذكرنا سابقًا، ثم يتم ربط هذه التضمينات الثلاث ونغذيّ الشبكة كما تحدثنا. تحتوي طبقات CNN على العديد من مُرشحات الالتفاف أحادية البعد، ولكل مُرشّح حجم نواة محدد 1+k=2c ويقوم بإجراء عملية الالتفاف والتنشيط RELU التالية:

$$x_{i,r}^{(l+1)} = \max\left(0, \left(\sum_{j=-c}^{c} w_{j,r}^{(l)} x_{i+j}^{(l)}\right) + b_r^{(l)}\right)$$

حيث تشير l إلى رقم الطبقة و r إلى رقم المُرشّح. يتم تطبيق كل مُرشّح على جميع مواقع الكلمات من i=1..n، ولهذا السبب يحسب كل مرشح تمثيل الكلمة i جمبًا إلى جمب مع الكلمات المجاورة c2 في سياقها. لاحظ أنني أقوم بإجبار حجم النواة على أن يكون فرديًا. حددت حجم خطوة الالتفاف على 1 وحددت حشوًا padding للمواقع left c وright c بأصفار. بهذه الحالة يكون خرج كل طبقة مُحاذي للدخل الأصلي بغية تحقيق هدف وضع العلامات على التسلسل.

طبقات BiLSTM المستخدمة هي من نوع Many-to-many، يتم إدخال التسلسل إلى كل طبقة، فتُخرج ترميزًا لهذا الدخل.

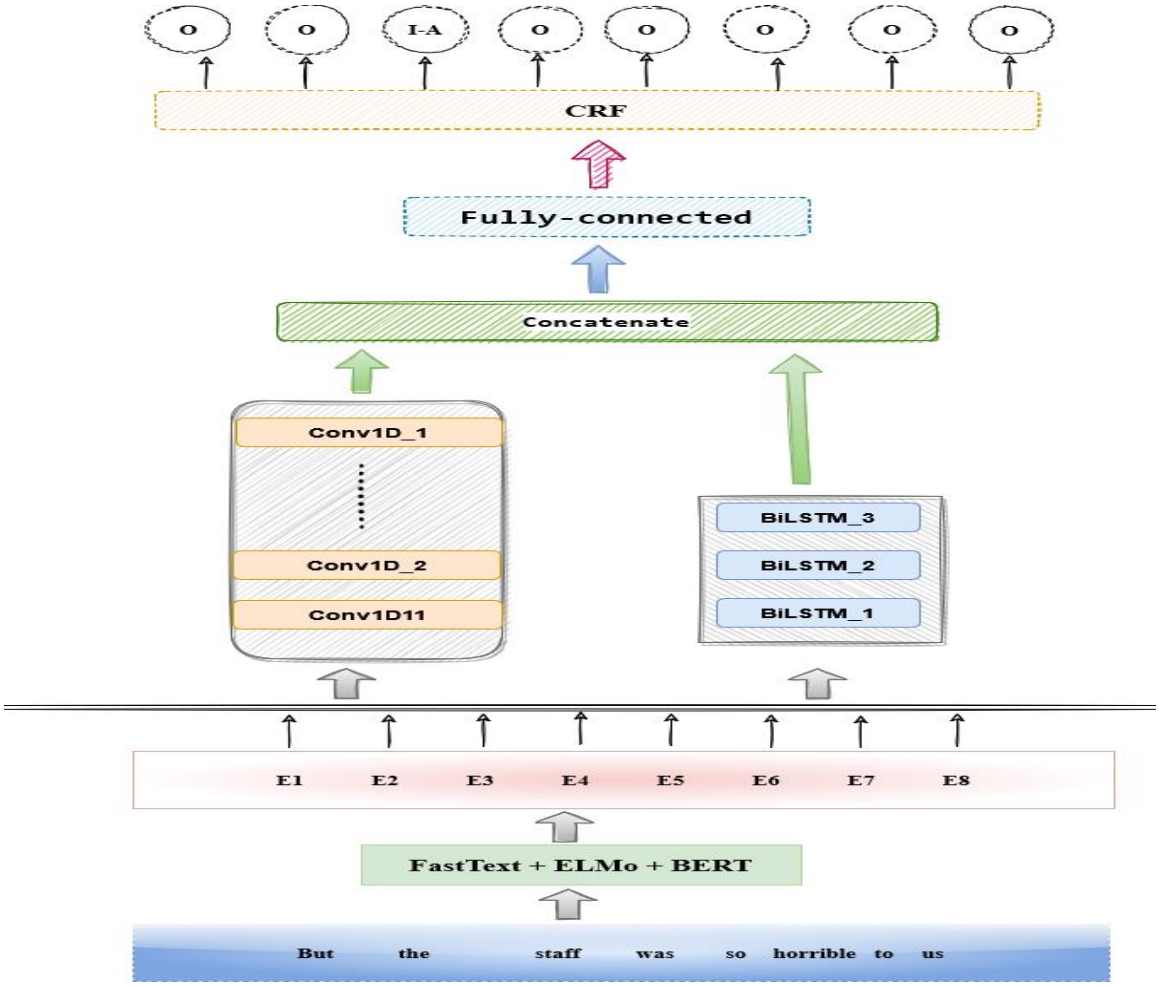
لاحظ أنني لا استخدم أي طبقات من نوع Maxpooling بعد طبقات الالتفاف، لأن نماذج وضع العلامات على التسلسل تحتاج إلى تمثيل جيد لكل موقع وهذا ما تعبث به هكذا طبقات، لذا فاستخدامها أمر غير مرغوب فيه.

بالنسبة للمعاملات العليا HyperParameters، استخدم 150 عينة تدريبية كبيانات تتبع لفعالية النموذج Validation لكي أضبط هذه المعاملات من خلالها. طبقات CNN الأولى والثانية تحويان 64 مرشح وحجم نواة 3 (c=1)، ومن الثالثة إلى الخامسة 128 مرشح وحجم نواة 3، السادسة 256 مرشح وحجم نواة 5 (c=2)، السابعة والثامنة 256 و 5، التاسعة والعاشرة والحادية عشرة 512 و 5.

طبقة BiLSTM الأولى تحتوي على 512 وحدة مخفية، الثانية والثالثة 256. طبقة الاتصال الكامل الأولى تحوي 512 وحدة مخفية، والثانية 256. طبقة الخرج CRF تحوي 3 وحدات في حالة تنسيق IOB وحدتين في حال تنسيق IO.

مُعدّل التسرب dropout بعد طبقة الاتصال الكامل الأولى 0.4 وبعد الثانية 0.3. مُعدّل التعلم لمُحسّن RMSprop وضعته على 0.0005 [46] لأن تدريب على هذه المهمة يميل إلى أن يكون غير مستقر. تجدر الإشارة إلى أن استخدام Adam بعطي نتائج أقل على هذه المهمة.

يتم استخدام دالة خسارة خاصة SigmoidFocalCrossEntropy [50] وهي دالة خسارة قوية جدًا في التعامل مع مهام التصنيف التي تتضمن عدم توازن في الفئات المُصنّفة. يعتمد على تقليل وزن الأصناف السهلة ويُركّز على الأصناف الصعبة. هذا ضروري في المهمة التي نتعامل معها فعدد الكلمات التي يتم تصنيفها على أنها جانب أقل بكثير من البيانات التي يتم تصنيفها على أنها ليست جانب. في هذه الحالة ستفرض هذه الدالة قيمة خسارة أكبر بكثير عندما يُخطئ النموذج في تصنيف الجوانب مقارنةً بالكلمات التي ليست جانب. عادةً ما تُستخدم هذه الدالة مع مهام الرؤية الحاسوبية في اكتشاف الكائنات (وجدت أن استخدامها هنا مفيدًا جدًا أيضًا) حيث يكون عدم التوازن بين فئة الخلفية والفئات الأخرى مرتفعًا للغاية.



الشكل (3-3) نموذج SE-EM

3-2- النتائج والتوصيات

أقارن النموذج مع أحدث الأساليب SOTA التي وصل إليها الباحثين الجدول (3-1). تم استخدام CRF التقليدية مع الميزات المصنوعة يدويًا. يستخدم HIS RD أيضًا CRF مع ميزات أجزاء الكلام POST و الكيان المسمى NER. نموذج LSTM كان ربما أول محاولة لاستخدام شبكة LSTM صافية. WDEmb حسّن نموذج CRF باستخدام تضمينات الكلمات وتضمينات السياق الخطية وتضمينات مسار التبعية.
DE-CNN يستخدم عدة طبقات CNN صافية مع زوج من التضمينات العامة والخاصة. CMLA عبارة عن شبكة انتباه مزدوجة متعددة الطبقات تقوم باستخراج الجوانب جمبًا إلى جمب مع تحليل المشاعر. HAST يعزز فكرة الاستخراج المشترك للجوانب مع المشاعر باستخدام شبكة الانتباه للتاريخ المتقطّع والتحول الانتقائي. RINANTE تشارك الميزات الموجودة في الطبقة السفلية من نموذج BiLSTM-CRF وتستخدم إشرافًا إضافيًا لتوسيع بيانات التدريب. النموذج BERT+BiSELF-CRF هو أفضل نموذج سابق على مجموعة بيانات SE14-R حيث يحقق 85.6 ومن خلال استخدام إشراف إضافي يتمثّل بشبكة المؤشر، يحقق النموذج تحسينًا إضافيًا 87.11. يتفوق النموذج SE-EM الذي اقترحه على النماذج السابقة في مجموعتي البيانات SE14-R و SE16-R، والتي تمثّل أحدث النتائج التي تحققت على هذه المهمة. يؤكد ذلك على مدى فاعلية استخدام مخطط التسمية IO في تحسين الأداء والاعتماد على تكديس التضمينات لإنتاج تمثيلات قوية جدًا للكلمات.

Method	SE14-R	SE14-L	SE16-R
CRF	79.72	72.77	66.69
HIS-RD	79.62	74.55	-
LSTM	82.01	75.71	70.35
WDEmb	84.97	75.16	-
NLANGP	-	-	72.34
DE-CNN	85.20	81.59	74.37
CMLA	85.29	77.80	-
HAST	85.61	79.52	73.61
RINANTE	84.06	84.06	
BERT+BiSELF-CRF	85.60	78.15	73.49
+Repositioning	87.11	81.90	75.56
BERT-BiSELF-CRF	85.60	80.15	75.64
+ Repositioning	87.11	82.68	77.51
BAT	81.50	85.57	-
PH-SUM	82.34	86.09*	
SE-EM (me)*	87.80*	84.5	80*

الجدول (3-1) مقارنة الأداء اعتمادًا على درجة F1

فيما يلي النقاط التي توصلت إليها والنقاط التي أوصي بها:

1. استخدام آلية التضمين التي تعتمد على دمج تضمينات سياقية وغير سياقية وعامة وخاصة تحقق أفضل النتائج.

2. استخدام التضمينات العامة أو الخاصة فقط يعطي أداءًا أقل.

3. استخدام طبقة CRF كطبقة خرج لا يؤثر على الأداء واستبدالها بطبقة FC تحوي خليتين مع تنشيط Softmax يُغني عنها. سبب ذلك هو أنه باستخدام التنسيق IO لن يعود النموذج قادرًا على رؤية التبعية التي سبق وتحدث عنها.

4. إن استخدام تضمينات سياقية خاصة بالمجال لا بد وأن يحسّن الأداء أكثر لكن ذلك يتطلب موارد حوسبية أكبر (يُفضل أن يكون لديك ذاكرة GPU بسعة أكبر من 14 جيجا).

5. يوجد في SE-EM ثلاث أنواع رئيسية من الأخطاء؛ يأتي أحد هذه الأخطاء من التسميات غير المتسقة. مثلًا، يتم تسمية كلمة A في بعض الأحيان على أنها جانب، وفي أحيان أخرى لا. النوع الآخر يأتي من الجوانب غير المرئية في بيانات الاختبار التي تتطلب دلالات كلمة الربط AND ليتم استخراجها. مثلًا، إذا كانت A جانب، وظهر لدينا A and B، فإن B يجب أن تعتبر جانب أيضًا، ولكن هذا لايحدث. النوع الأخير يأتي من مخطط التسمية الذي استخدمه IO، فالنموذج غير قادر على استخراج الجوانب التي تكون متتالية، أي إذا كان A جانب، وB جانب وظهرا معًا على التوالي، فإن النموذج سيعتبر أن A جانب وB جزءًا منه (يمكن حل هذه المشكلة بسهولة من خلال استخدام إشراف إضافي كما ذكرت سابقًا).

6. استخدام مخطط التسمية IO أكثر فعالية من استخدام مخطط IOB القياسي، ويحسّن الأداء بنسبة تصل إلى 2-3%.

7. اعتقد أن هذه المهمة تحتاج طفرة لتطوير النتائج أكثر، ربما من خلال إنتاج تضمينات أقوى أو استخدام بنى شبكات عصبية أكثر تعقيدًا. لابد أن نلاحظ أيضًا أن حجم البينات صغير قليلًا.

المراجع

1- Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers, pages 592–598.
2- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004, pages 168–177.
3- Bing Liu. 2012. Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers.
4- Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In CIKM ’06, pages 43–50.
5- Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: Modeling facets and opinions in weblogs. In WWW ’07, pages 171–180.
6- Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. Computational Linguistics, 37(1):9–27.
7- Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. arXiv preprint arXiv:1605.07843.
8- Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 388–397.
9- Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In EMNLP ’10, pages 1035–1045.
10- Lei Shu, Hu Xu, and Bing Liu. 2017. Lifelong learning crf for supervised aspect extraction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), volume 2, pages 148–154.
11- Maryna Chernyshevich. 2014. Ihs r&d belarus: Crossdomain extraction of product features using crf. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 309–313.
12- Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co extraction of aspect and opinion terms. In AAAI, pages 3316–3322.
13- Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2886–2892.

14- Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. Knowledge-Based Systems, 108:42–49.
15- Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. arXiv preprint arXiv:1603.06679.
16- Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. 1987. Occam's razor. Information processing letters, 24(6):377–380.
17- Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
18- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. 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), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
19- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
20- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8440–8451, Online. Association for Computational Linguistics.
21- Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Huang Zhongqiang, Fei Huang, and Kewei Tu. 2020b. More embeddings, better sequence labelers? In Findings of EMNLP, Online.
22- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.
23- Cicero D Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In Proceedings of the 31st international conference on machine learning (ICML-14), pages 1818–1826.
24- Pengfei Liu, Shafiq R. Joty, and Helen M. Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pages 1433–1443.
25- Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

26- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. arXiv preprint arXiv:1705.03122.
27- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In Advances in neural information processing systems, pages 649–657.
28- Erik Cambria and Amir Hussain. 2012. Sentic Computing Techniques, Tools, and Applications 2nd Edition.
29- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. Found. Trends Inf. Retr., 2:1– 135.
30- Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In HLT-EMNLP '05, pages 339–346.
31- Bo Wang and Houfeng Wang. 2008. Bootstrapping both product features and opinion words from Chinese customer reviews with cross-inducing. In IJCNLP '08, pages 289–295.
32- Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In ACL '08: HLT, pages 308–316.
33- Chenghua Lin and Yulan He. 2009. Joint sentiment/ topic model for sentiment analysis. In CIKM '09, pages 375–384.
34- Kang Liu, Liheng Xu, Yang Liu, and Jun Zhao. 2013. Opinion target extraction using partially-supervised word alignment model. In IJCAI '13, pages 2134–2140.
35- Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In ACL '13, pages 1643–1654.
36- John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML '01, pages 282–289.
37- Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. Neural computation, 1(2):270–280.
38- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543.
39- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.
40- Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems. Sowmya Vajjala & Bodhisattwa Majumder & Anuj Gupta & Harshit Surana. 17 June 2020
41- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pages 5998–6008.
42- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
43- Charles Sutton and Andrew McCallum. An Introduction to Conditional Random Fields. Vol.4,No.4 (2011)267–373.

44- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 27–35, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.
45- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation 8 OCT 2016.
46- Geoffrey Hinton with Nitish Srivastava Kevin Swersky 2012. rmsprop: Divide The gradient by a running average of its recent magnitude. 6 DEC 2021.
47- Vijay Krishnan and Vignesh Ganapathy 2005. Named Entity Recognition. December 16, 2005.
48- Zhenkai Wei, Yu Hong, Bowei Zou, Meng Cheng, Jianmin Yao 2020. Don’t Eclipse Your Arts Due to Small Discrepancies: Boundary Repositioning with a Pointer Network for Aspect Extraction. 5 July 2020.
49- Ronald J Williams. 1992. Simple statistical gradientfollowing algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256.
50- Tsung-Yi Lin Priya Goyal Ross Girshick Kaiming He Piotr Doll´ar. Focal Loss for Dense Object Detection. 7 FEB 2018.