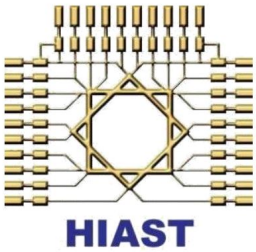
الجمهورية العربية السورية
المعهد العالي للعلوم التطبيقية والتكنولوجيا
قسم المعلوميات
العام الدراسي 2017/2018



نظام خبير لتقييم مقروئية نصوص اللغة الانكليزية وفق عدّة مستويات

مشروع السنة الرابعة

إعداد
فاروق حجابو

إشراف
د. غيداء ربداوي م. رياض سنبل

أيلول 2018

الملخص

تُعد القراءة نشاطاً مهماً نمارسه في حياتنا اليومية، سواء لغرض المطالعة أو قراءة الأخبار أو التعلم أو غيرها. يهدف هذا المشروع إلى بناء مُصنّفٍ قادرٍ على تقييم مقروئية النصوص المكتوبة باللغة الانكليزية، وتصنيفها وفق مستويات متدرجة (سهل، متوسط، صعب). يختلف عدد هذه المستويات ودرجة التباين بين صعوبتها تبعاً للمعطيات التي يتم استخدامها لبناء المصنّف. تتمحور منهجية العمل حول استخراج مجموعة من السمات من هذه النصوص: سمات تقليدية (مثل طول النص)، ومفرداتية (تعبّر عن تنوع المفردات المستخدمة)، ونحوية (تعكس الصياغة والتراكيب المستخدمة). بعد ذلك يجري استخدام خوارزميات تعلم الآلة (تم استخدام الـ SVM بشكل أساسي) لتدريب وبناء مصنّف باستخدام هذه السمات. جرى استخدام مجموعة المعطيات One Stop English Corpus، وكانت نسبة الصِحّة المحققة 80.83% وذلك باستخدام 30 سمة. علماً أن أعلى نسبة صِحّة في الأدبيات، تم الحصول عليها باستخدام هذه المعطيات هي 78.13% وذلك باستخدام 155 سمة [26]. كما جرى بناء مكتبة بلغة جافا لاستخراج السمات من النصوص، يمكن بسهولة استخدامها لاستخراج السمات التي تم تنجيزها من نص ما، أو توسيع هذه المكتبة بتعريف سمات جديدة.

Abstract

Reading is an important activity in our daily lives. We read the news, we read to learn, etc. This project aims to build a classifier to automatically assess text readability for texts written in English. By classifying them into different levels (easy, medium, and hard). Noting that, the number of levels and the variance of their difficulties depend on the data used to build the classifier. The methodology is to extract certain set of features from these texts. Traditional features (e.g. text length), lexical features (measure vocabulary being used), and syntactic features (measure sentence complexity). Then using machine learning algorithms (SVM mainly) to train and build a classifier using these features. Using One Stop English Corpus as the dataset, we achieved an accuracy of 80.83% (using 30 feature). Given that the best achieved accuracy on this dataset is 78.13% using 155 feature [26]. Also, we implemented a library in java to extract features from texts. It can be easily used to extract the features we implemented from a given text. It also can be extended by implementing new features.

شكر

أتوجه بجزيل الشكر إلى أسرة المعهد العالي كمؤسسة تعليمية راقية. وأخص بالشكر كادر قسم المعلوميات.

وأخص بالذكر المشرفين على هذا المشروع، الدكتورة غيداء ربداوي. والمهندس رياض سنبل الذي أعطاني جزء كبير من وقته ووجهني في جميع خطوات هذا المشروع.

وأحب أن أوجه شكر إلى مجموعة الباحثين في هذا المجال والذين جعلو أبحاثهم متاحة على الانترنت. وأخص بالذكر السيد Sowmya Vajjala الذي ساهم في تجميع المعطيات وجعلها متوفرة مجاناً على الانترنت بالإضافة إلى مجموعة من الأكواد البرمجية المفيدة.

المحتويات

قائمة الأشكال

قائمة الجداول

الاختصارات

SVM	Support Vector Machine
SMO	Sequential Minimal Optimization
NLP	Natural Language Processing
OSE	One Stop English Corpus

المصطلحات

Artificial Intelligence	الذكاء الصنعي
Machine Learning	تعلّم الآلة
Natural Language Processing	معالجة اللغات الطبيعية
Supervised Learning	التعلم المشرف عليه
Unsupervised Learning	التعلم غير المشرف عليه
Semi-Supervised Learning	التعلم المشرف عليه جزئياً
Reinforcement Learning	التعلم بالتعزيز
Classification	التصنيف
Regression	الانحدار
Training Set	مجموعة التدريب
Test Set	مجموعة الاختبار
Training Instance	مثال تدريب
Accuracy	الصِحَّة
Precision	الدقّة
Recall	الإرجاع
Clustering	العنقدة
Features	سمات
Feature Extraction	استخراج السمات
Regularization	التنظيم
Kernel	نواة

Linear Kernel	النواة الخطية
Polynomial Kernel	النواة الحدودية
Gaussian Kernel	النواة الغاوسية
Hyperparameter	باراميتر فوقي
Classifier	مُصنّف
Morphological Analysis	التحليل الصرفي
Tokenization	التقطيع
Stemming	التجذير
Chunking	التقطيع الجمل
Parsing Tree	الشجرة النحوية
Dependency Parsing	تحليل التّبعية
Dependency Graph	بيان التبعية
Anaphora	الإحالة
Design Patterns	الأنماط التصميمة
Confusion Matrix	مصفوفة الخطأ

الفصل الأول

التعريف بالمشروع

يُمهّد هذا الفصل للمشروع، حيث يُبيّن فكرة المشروع وأهميته والأهداف المرجوّة منه. ويذكر المتطلبات الوظيفية وغير الوظيفية للمشروع. ويوضح الخطة الزمنية لتنفيذ المشروع.

1.1 مقدمة

تلعب القراءة دوراً مهماً في تعلم لغةٍ جديدةٍ أو اكتساب معارف ومعلومات جديدة حول موضوع معيّن. بالتالي فإن أي مسببات للصعوبة أثناء عملية القراءة ستؤثر سلباً على عملية التعلم واكتساب المعارف. اهتم الباحثون بالأسباب التي تؤدي إلى صعوبة قراءة النصوص وتأثيراتها على القُرّاء حيث تمت دراسة الخصائص اللغوية التي تسبب صعوبة قراءة النصوص؛ سواء على مستوى المفردات أو على مستوى التراكيب والجمل. وأجريت عدّة دراسات تحاول بناء نماذج لتقييم مقروئية النصوص Text Readability Assessment بشكل آلي، وهو ما سنحاول العمل عليه في هذا المشروع.

2.1 أهمية المشروع وتطبيقاته

تتنوع تطبيقات أنظمة تقييم مقروئية النصوص لتشمل أنواعاً مختلقة من المستخدمين. فيمكن للاساتذة استخدامه لمساعدتهم في اختيار نصوص مناسبة لطلابهم سواء أثناء الجلسات التعلمية أو في الاختبارات، وخاصةً أساتذة تعليم اللغات. هذا التطبيق سيساعد الطلاب على اختيار ما يناسبهم أثناء عملية دراستهم لموضوع معين أو قراءة مقالات حول مجال ما خاصةً بوجود معلومات هائلة متاحة على الانترنت،. وبعيداً عن سياق الأمور التعليمية، يمكن لتحليل صعوبة نص أن تكون مناسبة ولازمة في عدّة سيناريوهات مثل تحليل النصوص القانونية والقضائية. أيضاً يمكن للكُتَّاب الاستفادة من هكذا تطبيق أثناء عملية كتابتهم، سواء كتابة مقال علمي أو مقال صحفي أو خبر أو غيرها.

ولإعطاء تطبيقات ملموسة بشكل أكثر، سنتحدث لاحقاً عن عدد من النصوص التي تم استخدامها ضمن المشروع، حيث يختار مجموعة أساتذة نص معيّن، ويعيدون صياغته إلى ثلاثة نصوص بما يناسب طلاب من ثلاثة مستويات متدرجة. مع المحافظة على فحوى النص بأكبر شكل ممكن، ولكن صياغته ستختلف لتناسب ثلاث مستويات من الطلاب. فوجود هذا التطبيق سيساعدهم في معرفة ما إذا كانت صياغتهم مناسبة أم أنهم يحتاجون إلى تبسيطه بشكل أكبر.

يمكن استخدام هذا التطبيق لمساعدة أساتذة اللغة الإنكليزية. سواء في المعهد العالي أو المدارس أو غيرها. فعادةً يوجد قسم في امتحان اللغة لتقييم قدرات الطالب على فهم نص جديد في اللغة الإنكليزية reading comprehension. إن ما يقوم به الاساتذة أحياناً هو اختيار نص من الكتاب نفسه لم يتم عرضه مسبقاً على الطلاب. أو اختيار نص من الانترنت، فيمكن أن تصبح هذه العملية أكثر سهولة باستخدام هذا التطبيق ليكون هذا النص أكثر ملائمة لمستوى الطلاب، وبالتالي أفضل لتقييم الطلاب بشكل سليم وعادل وأكثر موضوعيّة.

3.1 المتطلبات

نسرد فيما يلي المتطلبات الوظيفية وغير الوظيفية للمشروع.

1.3.1 المتطلبات الوظيفية

1. بناء تطبيق لتقييم سهولة قراءة نص مكتوب باللغة الانكليزية. يتمتع بالمزايا التالية:

(ا) المصنِّف المستخدم يتعلق بالمعطيات المستخدمة للتدريب. (من ناحية المستويات التي يتم تصنيف صعوبة النص وفقها، وعددها، والتفاوت بينها).

(ب) يُتيح التطبيق للمستخدم تجريب عدّة مُصنفات لتصنيف نص مُدخَل.

(ج) تنجيز مُصنّف واحد على الأقل.

2. بناء مكتبة برمجية تساعد على استخراج السمات لنص أو مجموعة نصوص.

2.3.1 المتطلبات غير الوظيفية

1. الفعاليّة والوثوقية. يجب أن يحقق النظام نسبة صحّة مقبولة (تتجاوز 75%).
2. الكفاءة. يستغرق التطبيق وقت بسيط لتصنيف نص معيّن (لا يتعدا 10 ثواني لنص طوله أقل من 1000 كلمة).
3. قابلية التوسّع. يمكن إضافة مصنفات جديدة باستخدام المعطيات ذاتها أو باستخدام معطيات جديدة.
4. يجب أن تحقق مكتبة استخراج السمات ما يلي:

(ا) قابلية التوسّع. يمكن لمستخدم المكتبة تنجيز سمات جديدة.

(ب) سهولة الاستخدام. يمكن لمستخدم المكتبة استخدام السمات المنجّزة بشكل مسبق بسهولة والتركيب بينها.

4.1 الخطة الزمنية للمشروع

لم تكن هناك خطة زمنية واضحة لتنفيذ المشروع. فقد تم العمل على مراحل جزئية، كل مرحلة استغرقت حوالي أسبوع. وكان العمل كالتالي:

1. الأسبوع الأول 15/7: دراسة نظرية (محو أميّة) حول مفاهيم تعلم الآلة وخوارزمياتها المختلفة. وتجميع ودراسة بعض الأبحاث النظرية حول تقييم مقروئية النصوص.
2. الأسبوع الثاني 22/7: دراسة المنهجيات المستخدمة ضمن الأوراق البحثية حول موضوع المشروع. وتأمين مجموعة المعطيات. وتحضير قائمة بأهم السمات المستخدمة في هذه الأبحاث.

3. الأسبوع الثالث 22/7: تنجيز نموذج أولي prototype. تنظيف المعطيات. تنجيز عملية استخراج السمات من النصوص. تطبيق عدد من خوارزميات تعلم الآلة للحصول على نتائج أولية.
4. الأسبوع الرابع 5/8: توقف العمل بسبب المشاركة في معسكر تدريبي للتحضير للمسابقة البرمجية السورية ACM.
5. الأسبوع الخامس 12/8: تحسين وإعادة هيكلة النظام. تحسين التنجيز السابق بإعادة تصميم الكود البرمجي. وتطبيق خوارزميات جديدة لتحسين النتائج المرحلية.
6. الأسبوع السادس 19/8: توقف العمل بسبب عطلة عيد الأضحى.
7. الأسبوع السابع 26/8: بناء واجهة التطبيق النهائي. البدء بتحضير التقرير النهائي للمشروع.
8. الأسبوع الثامن 2/9: استكمال كتابة التقرير. تحضير العرض العملي.
9. الأسبوع التاسع 9/9: استكمال كتابة التقرير. تحضير العرض التقديمي النهائي.

الفصل الثاني

الدراسة المرجعية

يبيّن هذا الفصل أهم المفاهيم والتجارب المرتبطة بهذا المشروع. يبدأ بتقديم مفاهيم تعلّم الآلة ومراحلها المختلفة والمعايير المعتمدة لتقييمها. ثم يقدّم المفاهيم والمراحل الأساسية لمعالجة اللغات الطبيعية. وأخيراً نسرد أهم الأبحاث العلمية المتعلقة بهذا المشروع، ونوضح المنهجيات المتبعة في كل منها.

1.2 تعلم الآلة

تعلم الآلة Machine Learning هو أحد فروع من الذكاء الصنعي Artificial Intelligence. يُقصد بتعلم الآلة مجموعة الأدوات والمفاهيم والمنهجيات المستخدمة لبرمجة الحواسيب بطريقة تسمح لهذه الحواسيب بالتعلم من المعطيات [24].

وقد عرّف آرثر ساموبل تعلم الآلة بشكل أكثر عمومية بأنّه [24]:

"Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed."
–Arthur Samuel, 1959

أما توم ميتشل فقد أعطى تعريفاً أكثر تحديداً [24]:

"A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."
–Tom Mitchell, 1997

على سبيل المثال، معظم الأنظمة التي تقوم بفلترة الإيميلات الإلكترونية إلى إيميلات مؤذية spam وإيميلات غير مؤذية non-spam، تستخدم منهجيات تعلم الآلة. تقوم هذه الأنظمة بتعلم طريقة التمييز بين هذين النوعين من الإيميلات باستخدام عدد كبير من الأمثلة والمعطيات المصنفة مسبقاً. نسمي هذه المجموعة من الأمثلة بمجموعة التدريب Training Set، وكل مثال منها نسميه مثال تدريب Training Instance.

في هذه الحالة، المهمة T هي تصنيف الإيميلات الجديدة إلى إيميلات مؤذية وإيميلات غير مؤذية، الخبرة E هي مجموعة مجموعة التدريب، ومؤشر قياس الأداء P يمكن تعريفه بعدّة طرق؛ فمثلاً يمكننا استخدام نسبة عدد الإيميلات التي تم تصنيفها بشكل صحيح إلى عدد الإيميلات الكلي (هذا المعيار يسمى الصِحَّة Accuracy كم سنرى لاحقاً).

1.1.2 المنهجيات العامة لتعلم الآلة

يمكن تصنيف أنظمة تعلم الآلة وفق عدّة معايير. ولعّل التصنيف الأكثر شهرة هو التصنيف الذي يعتمد على آلية التدريب، وهو كالتالي:

- التعلم المشرف عليه Supervised Learning: عندما تكون الأمثلة التدريبية متاحة مع الخرج label المرتبط بها. كما في مثال تصنيف الإيميلات المطروح سابقاً. حيث أن مجموعة التدريب هي مجموعة كبيرة من الإيميلات المصنفة مسبقاً من قبل البشر إلى إيميلات مؤذية وإيميلات غير مؤذية.
- التعلم غير المشرف عليه Unsupervised Learning: عندما تكون مجموعة التدريب موجودة ولكنها غير مصنفة unlabeled أو غير مرتبطة بخرج معيّن. على سبيل المثال، قد ترغب شركة في تصنيف زبائنها إلى عدّة مستويات، زبائن من الدرجة الأولى، زبائن من الدرجة الثانية، وهكذا. فيمكن استخدام تعلم الآلة لاكتشاف بعض الأنماط الموجودة في معطيات الزبائن واكتشاف هكذا تصنيف. وهذا ما يُعرف بالعنقدة Clustering.
- التعلم المشرف عليه جزئياً Semi-Supervised Learning: وهي حالة وسيطة بين التصنيفين السابقين. تكون فيها بعض أمثلة التدريب مرتبطة بخرج معيّن (غالباً تشكل النسبة الصغيرة)، وتكون باقي الأمثلة

غير مرتبطة بخرج. تنطبق هذه الحالة على مثال تصنيف الإيميلات في حال لم تكن جميع مجموعة التدريب مصنفة بشكل مسبق.

- التعلم بالتعزيز Reinforcement Learning: وهي الحالة التي يتخاطب فيها النظام مع البيئة المحيطة به. تقدم له هذه البيئة نتائج feedback بناءً على أفعاله. تعد الألعاب أشهر التطبيقات التي يستخدم فيها هذا النوع من التعلم، حيث يقوم النظام بمجموعة من الأفعال actions ضمن بيئة اللعبة، وبناءً على النتائج (تحسّن نتيجته أو انخفاضها) يغيّر أفعاله اللاحقة.

وعلى وجه الخصوص يمكن تصنيف التعلم المشرف عليه بحسب نوع الخرج المرتبط بمجموعة التدريب. تصنّف بشكل أساسي كالتالي:

- التصنيف Classification: يكون الخرج المرتبط بكل مثال تدريب هو صف class محدد من مجموعة صفوف. عدد هذه الصفوف قد يكون 2، 3، إلخ. في مثال تصنيف الإيميلات السابق، عدد الصفوف هو 2، حيث أن كل مثال تدريب (إيميل معيّن من مجموعة التدريب) هو إمّا مؤذي أو غير مؤذي.
- الانحدار Regression: يكون الخرج المرتبط بكل مثال تدريب هو عدد حقيقي. مثل مسألة التنبؤ بسعر منزل بمعرفة معلومات عنه مثل مساحته، عدد الغرف، إلخ.

2.1.2 المراحل اللازمة لتطبيق تعلم الآلة

يبدأ العمل في الأنظمة المعتمدة على منهجيات تعلم الآلة بمرحلة تجميع المعطيات، ومرحلة تنظيفها. تتم عملية تجميع المعطيات بحسب التطبيق. فمثلاً قد تكون المعطيات هي نتيجة استبيانات، أو إحصائيات، أو تم الحصول عليها من مواقع إلكترونية، إلخ. بينما تهدف مرحلة تنظيف المعطيات إلى التأكد سلامة هذه المعطيات قبل استخدامها. وقد تتم هذه العملية بشكل يدوي أو بشكل مؤتمت وذلك بحسب مصدر المعطيات ونظافتها.

ذكرنا في مثال تصنيف الإيميلات الالكترونية أن مجموعة التدريب هي مجموعة من الإيميلات المصنفة بشكل مسبق إلى إيميلات مؤذية وإيميلات غير مؤذية. يمكن أن نسأل هنا: ما هو تحديداً دخل عملية التعلم؟ أي كيف سنعبّر عن الإيميل؟ بالطبع يمكن اعتبار الإيميل كنص؛ كمجموعة من الكلمات والرموز. ولكن كما سنرى لاحقاً، من الصعب على معظم خوارزميات تعلم الآلة التعامل مع نص خام. ولذلك هناك مرحلة تسبق مرحلة تنفيذ خوارزميات تعلم الآلة وهي مرحلة تحويل النص إلى ما يسمى بالسمات Features.

فمثلاً يمكن أن نعبّر عن نص الإيميل بسماته، مثل عدد الكلمات، عدد الجمل، تواتر وجود كلمات مفتاحية

محددة، إلخ. أي أننا نتعامل مع الإيميل كشعاع من السمات feature vector وهذا أمر مناسب جداً للعديد من خوارزميات تعلم الآلة. إن السمات التي ذكرناها في مثالنا هي سمات عددية numerical features، بشكل عام يمكن أن تكون السمات هي سمات نصية string features أو سمات صنفية categorical features، إلخ. ويمكن أيضاً تنميط السمات بشكل مختلف أو أكثر دقة مثل تصنيف السمات العددية إلى سمات مستمرة continuous features وسمات متقطعة discrete features. وتعود طريقة توصيف السمات إلى التطبيق أو خوارزميات تعلم الآلة المستخدمة. تسمى هذه المرحلة بمرحلة استخراج السمات Feature Extraction.

3.1.2 بعض خوارزميات تعلم الآلة

كما رأينا في الفقرة 1.1.2، هناك العديد من أصناف المسائل الممكن حلها باستخدام تعلم الآلة. تصنف خوارزميات تعلم الآلة تبعاً لصنف المسألة التي تقوم بحلها. فمثلاً يمكن استخدام الانحدار الخطي Linear Regression لحل مسائل الانحدار [24]. أو استخدام خوارزمية K-Means Clustering لحل مسائل العنقدة [24]. سنمهد في هذه الفقرة للخوارزمية المستخدمة في هذا المشروع الـ SVM، وهي من أشهر خوارزميات تعلم الآلة.

خوارزمية الـ SVM

إن كلمة SVM هي اختصار لـ Support Vector Machine. وهي خوارزمية تصنيف شهيرة وواسعة الاستخدام في تطبيقات تعلم الآلة. تعتبر خوارزمية قوية حيث أنها تستند على أساس رياضي متين، ولها عدد من الخصائص المهمّة. يمكن تقديم هذه الخوارزمية بعدّة طرق. سنقدمها بطرح مسألة الأمثلة التي تقوم بحلها.

بدايةً لنفرض أن مسألتنا هي مسألة تصنيف وعدد الصفوف هو 2. نرمز بـ $(x^{(i)}, y^{(i)})_{1\leq i\leq m}$ إلى مجموعة التدريب. حيث m عدد مجموعة التدريب. ويكون المثال التدريبي رقم i، له الصف $y^{(i)}$. مع كون $y^{(i)} = +1$ في حال الصف الأول، و $y^{(i)} = -1$ في حال الصف الثاني. وإن $x^{(i)}$ هو شعاع عددي بـ $n+1$ بُعد أي $x^{(i)} \in \mathbb{R}^{n+1}$. وهو ما سميناه شعاع السمات في الفقرة 2.1.2. أي هنا لدينا n سمة، حيث لتبسيط العلاقات الرياضية نضيف $x_0^{(i)} = 1$.

النموذج المطروح في خوارزمية الـ SVM، هو تعريف تابع $f : \mathbb{R}^{n+1} \to \{-1, +1\}$. حيث أننا نقول أنه لأجل عيّنة ما (x, y)، فإنها تنتمي إلى الصف الأول في حال كان $f(x) \geq 0$، وتنتمي إلى الصف الثاني في

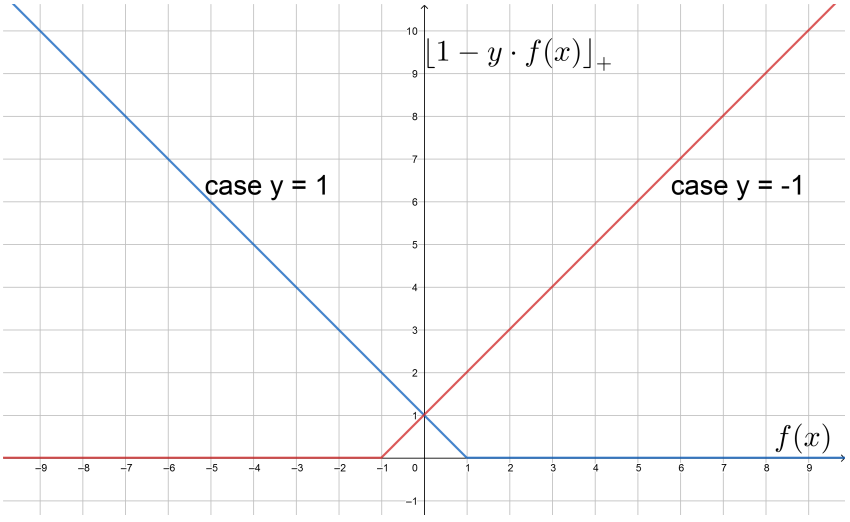
حال $f(x) < 0$. سنأخذ مبدئياً للتبسيط التابع f بالشكل $f(x) = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n$ حيث $\theta = (\theta_j)_{0 \le j \le n}$ هي البارامترات التي يمكن تغييرها.

مسألة الأمثلة التي نريد حلها هي:

$$\underset{\theta \in \mathbb{R}^{n+1}}{\operatorname{argmin}} \left(\|\theta\|_2^2 + C \cdot \sum_{i=1}^{m} \left\lfloor 1 - y^{(i)} f(x^{(i)}) \right\rfloor_+ \right) \quad (1)$$
$$\text{where} f(x^{(i)}) = \sum_{j=0}^{n} \theta_j x_j^{(i)}$$

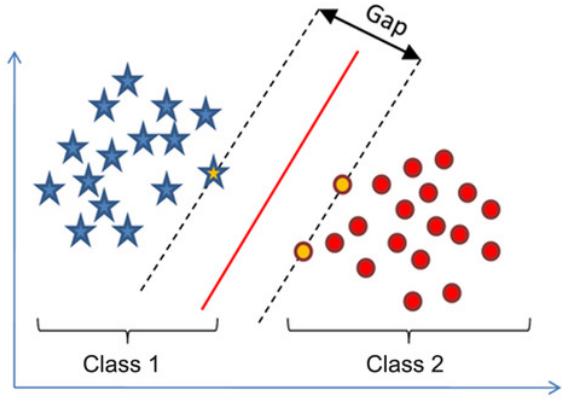
حيث أن التابع $\lfloor \cdot \rfloor_+$ هو تابع الجزء الموجب؛ أي $\lfloor z \rfloor_+ = \max(z, 0)$. و $\|\cdot\|_2$ هو النظيم الإقليدي؛ أي $\|\theta\|_2^2 = \sum_{j=0}^{n} \theta_j^2$. والبارامتر C هو معامل وزن، يحدد مدى التفضيل والمساومة بين الحديّن الأول والثاني في المعادلة. وهو بارامتر فوقي Hyperparameter أي يجب تحديده قبل البدء بحل مسألة الأمثلة، وإن تغييره يغيير حل المسألة.

إن الحد الأول $\|\theta\|_2^2$ في المعادلة 1 هو للتنظيم Regularization. هذا الحد يضبط قيم البارميتر θ ويمنعها من أن تأخذ قيم كبيرة. الحد الثاني يمثل مجموع قيمة الخطأ الحاصل في كل مثال تدريب من مجموعة التدريب. حيث أن الخطأ الحاصل في عيّنة ما (x, y) هو $\lfloor 1 - y \cdot f(x) \rfloor_+$. يمكن تأمل صفات هذا الخطأ من خلال الشكل 1.2. حيث نلاحظ مثلاً في حالة $y = 1$ أن الخطأ يساوي الصفر عندما $f(x) \ge 1$ وأنه يتزايد بشكل خطي كلما أبتعدت قيمة $f(x)$ عن 1 بالاتجاه الخاطئ.



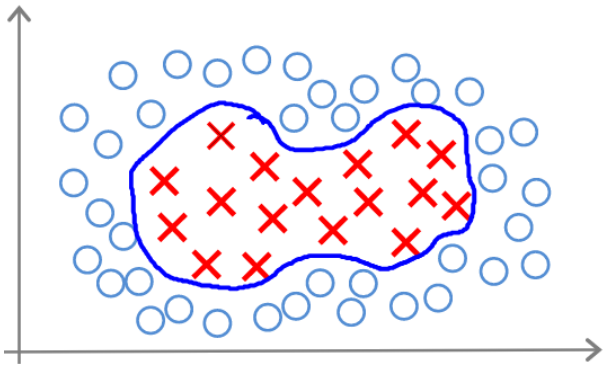
الشكل 1.2: الخطأ في العينة الواحدة في نموذج الـ SVM.

يمكن البرهان على أنه في حالة كون المعطيات قابلة للفصل بخط مستقيم، فإن حل مسألة الأمثلة سيعطي المستقيم f الذي يحقق أكبر هامش ممكن؛ أي اذا قمنا بحساب البعد بين كل نقطة وهذا المستقيم، فإن أصغر بعد سيكون أكبر ما يمكن، وهذا ما يوضحة الشكل 2.2.



الشكل 2.2: مستقيم يفصل صفين بهامش أعظمي.

ولكن أيضاً يمكننا اختيار تابع غير خطي. هذا مفيد مثلاً في حال كان شكل مجموعة التدريب مثلما في الشكل 3.2. إذ يوجد أسلوب يسمى بالـ Kernel Trick، يسمح لنا بفعل هذا. ينص هذا الأسلوب على



الشكل 3.2: مجموعة التدريب غير قابلة للفصل باستخدام مستقيم.

تعريف f بالشكل $f(x) = \sum_{i=1}^{m} \theta_i K(x, x^{(i)}) + \theta_0$، ثمّ حل مسألة الأمثلة السابقة ذاتها. نلاحظ هنا أنه لدينا $m+1$ باراميتر عوض الـ $n+1$ باراميتر في الحالة السابقة. و يسمى التابع K بالنواة Kernel. فمثلاً اختيار $K(u, v) = \sum_{j=1}^{n} u_j v_j$ يؤدي إلى حل مكافئ لحل المسألة الموضحة في المعادلة 1. تسمى هذه النواة بالنواة الخطية Linear Kernel.

إنّ أشهر النوى المستخدمة عادةً هي:

النواة الخطية	Linear Kernel	$K(u,v) = u^T v$
النواة الحدودية	Polynomial Kernel	$K(u,v) = (u^T v + r)^d$
النواة الغاوسية	Gaussian Kernel	$K(u,v) = \exp\left(-\gamma \left\|u - v\right\|_2^2\right)$

إذ أن الرمز u^T يرمز إلى المنقول وتحديداً فإن $u^T = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}^T = (u_1, \ldots, u_n)$. تسمى النواة الغاوسية أيضاً بـ Radial Basis Function (RBF) Kernel. وننوه أن البارمترات المذكورة r, d, γ هي بارامترات فوقية.

لحل مسألة الأمثلة المطروحة، توجد العديد من الخوارزميات. هذا النوع من المسائل، ومسائل الأمثلة بشكل عام هو فرع مدروس بشكل جيد في الرياضيات تحت اسم Mathematical Optimization. فتوجد العديد من الخوارزميات المستخدمة لحل مسألة الأمثلة المطروحة. من أشهرها هي خوارزمية Sequential Minimal Optimization (SMO). يتطلب شرحها الخوض في كثير من التفاصيل الرياضية وهو خارج نطاق هذا المشروع.

آخر ما يجب ذكره، هو الأسلوب المستخدم للتصنيف في حال وجود أكثر من صفين. فكما رأينا، إن خوارزمية الـ SVM مبنية للتصنيف بين صفين فقط، وهذا ما يسمى بالتصنيف الثنائي binary classification. توجد عدّة أساليب تسمح باستخدام خوارزميات التصنيف الثنائي لإجراء تصنيف عندما يكون عدد الصفوف هو $k > 2$. سنستخدم أسلوب يسمى بـ One-versus-One multi-class classification. الفكرة كالتالي، لأجل كل صفين مختلفين، نهمل بقيّة الصفوف، ونبني مصنّف للتمييز بين هذين الصفين. أي نقوم ببناء $k(k-1)/2$ مصنف. الآن لأجل عينة جديدة، نقوم بتطبيق هذه المصنفات جميعها، ونأخذ الصف الذي تم اختياره أكبر عدد من المرات.

4.1.2 معايير التقييم

تختلف معايير تقييم صحة نماذج تعلم الآلة باختلاف نوع المسائل التي تقوم بحلها. سنتحدث في هذه الفقرة عن أهم معايير التقييم المستخدمة في مسائل التصنيف.

بدايةً لنضع بعض الرموز لتبسيط العلاقات الرياضية وتوضيح الأفكار. كما تحدثنا سابقاً عن مجموعة التدريب، من المعتاد أن توجد معطيات أخرى مستقلة عن مجموعة التدريب تسمى بمجموعة الاختبار Test Set. حيث أنه بعد الحصول على النموذج الناتج من خوارزمية تعلم الآلة بتدريبه على مجموعة التدريب، يتم اختبار هذا النموذج على مجموعة الاختبار. سنرمز لها بـ TS. سنرمز لمجموعة عناصرها بـ (x_i, y_i)، حيث x_i هو شعاع

السمات، y_i هو الصف الموافق. وسنرمز بـ $\hat{y}_i$ للصف الذي تنبأت به خوارزمية تعلم الآلة المستخدمة والتي نريد تقييمها. وسنستخدم الرمز $|\cdot|$ لعدد عناصر مجموعة ما. فمثلاً إن $|y_i = c|$ هو عدد العناصر من TS التي لها الصف c.

الصِحّة Accuracy هي المعيار الأشهر. فهي نسبة العينات التي تم تصنيفها بشكل صحيح. أي:

$$\text{Accuracy} = \frac{|\hat{y}_i = y_i|}{|\text{TS}|}$$

إنّ هذا المعيار غير كافي للتعبير عن مدى قوة النموذج الناتج. لنتأمل مثال تكون فيه مجموعة التدريب فيها صفين فقط. نسبة ورود الصف الأول هو 1%، مثل حالة تشخيص مرض نادر. فبإمكاننا بسهولة الحصول على نموذج بدقة 99%. هذا النموذج يتنبأ دائماً بالصف الثاني؛ فلكون ورود عينات تنتمي للصف الأول نادر جداً تكون صحة هذا النموذج عالية. ولكن من الواضح أن هذا النموذج غير مجدي. النقاش السابق يدفع لتحديد معايير أخرى للتقيم.

الدقّة Precision هي معيار يعبر عن دقّة تصنيف صف معيّن. دقّة تصنيف الصف c هي نسبة العينات التي صنفت بشكل صحيح في الصف c من بين جميع العينات التي صنفت بالصف c. أي:

$$\text{Precision for class c} = \frac{|\hat{y}_i = c \wedge y_i = c|}{|\hat{y}_i = c|}$$

الإرجاع Recall هو معيار يعبر عن مدى استرجاعنا لعينات من صف معيّن. معيار الإرجاع للصف c هو نسبة العينات التي صنفت بشكل صحيح في الصف c من بين جميع العينات التي هي ضمن الصف c فعلاً. أي:

$$\text{Recall for class c} = \frac{|\hat{y}_i = c \wedge y_i = c|}{|y_i = c|}$$

المعيار الأخير الذي سنتحدث عنه يسمى بـ F1-score. ينتج من حاجتنا إلى الاعتماد على قيمة عددية واحدة فقط لمقارنة نموذجين معاً. وهو معيار يجمع بين الدقّة والإرجاع. النموذج المقترح للجمع بينهما هو:

$$\text{F1} - \text{score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

سنرمز لهذا المعيار اختصاراً بـ F-score. حيث أن الرقم 1 في اسمه يدل على أننا نعطي للدقّة والإرجاع نفس

الأهمية. فهذا المعيار حالة خاصة من معيار أعم يسمح بإعطاء أهمية أكبر للدقّة على الإرجاع وبالعكس، ولكن لن نتحدث عنه.

2.2 معالجة اللغات الطبيعية

معالجة اللغات الطبيعية Natural Language Processing (NLP) هو المجال الذي يدرس الآليات التي تسمح للحواسيب والآلات بفهم ومعالجة اللغات الطبيعية مثل اللغة العربية والإنكليزية وغيرهما [25]. ويعتبر مجال مهم في الذكاء الصنعي. يتقاطع هذا المجال مع العديد من المجالات منها علم اللسانيات وتعلم الآلة وغيرهما. سنتحدث في هذه الفقرة بشكل بسيط عن أهم المراحل في معالجة اللغات الطبيعية.

- التحليل الصرفي Morphological Analysis وهو المرحلة التي يجري فيها تحليل الكلمة إلى مكوناتها الأساسية. يُنفّذ هذا التحليل على مستوى الكلمة دون النظر إلى السياق. بشكل أساسي هناك مرحليتن لهذا التحليل. التقطيع Tokenization والتجذير Stemming.

  خرج مرحلة التقطيع هو الرموز التي تكون الجملة Tokens. أي مثلاً إنّ الجملة

  Google inc. is huge.

  سيتم تقسيمها إلى خمس رموز وهي:

  {Google | inc. | is | huge | .}

  لاحظ أن النقطة الأخيرة تعتبر رمز منفصل بينما النقطة في الكلمة .inc ليست رمز منفصل.

  خرج مرحلة التجذير هو جذر الكلمة، بالإضافة إلى السوابق واللواحق، ومعلومات أخرى تختلف باختلاف اللغة. مثلاً إن جذر الكلمات fish, fishing, fished, fisher هو fish. واللواحق هي ing, ed, er على الترتيب.

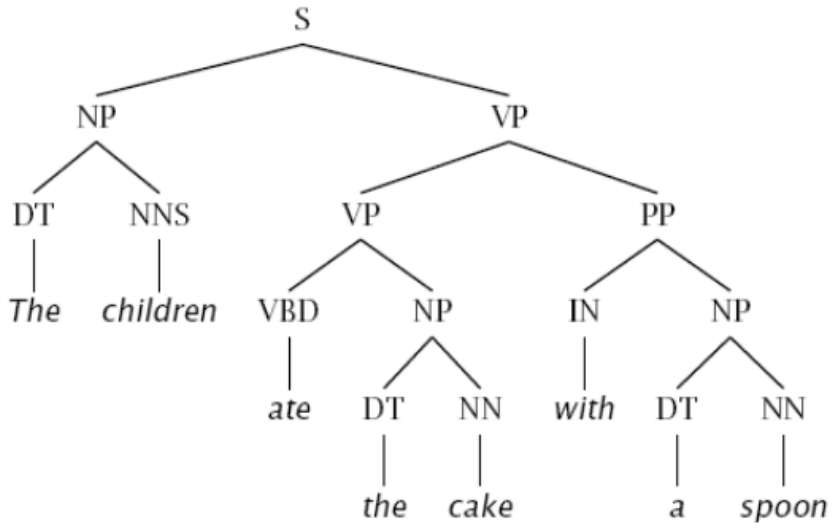
- تحديد أنماط الكلمات Part-of-Speech Tagging وهو عملية إسناد الأنماط النحوية الملائمة لكل كلمة من كلمات الجملة. دخل هذه المرحلة عادةً يكون كلمة ضمن سياق محدد (ضمن جملة). الخرج الناتج يكون النمط النحوي لهذه الكلمة (اسم، فعل، صفة، إلخ).

- التقطيع الجمل Chunking وهو تقسيم الجملة إلى عبارات أصغر؛ أي عبارات اسمية، أو عبارات فعلية، إلخ. يوضح الشكل 4.2 مثال على ذلك.

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September]

الشكل 4.2: عمليّة التقطيع الجمل للجملة He reckons the current account deficit will narrow to only # 1.8 billion in September. تم تقسيم هذه الجملة إلى جمل فعلية Verb Phrase (VP)، وجمل اسمية Noun Phrase (NP)، وعبارات جر Prepositional Phrase (PP).

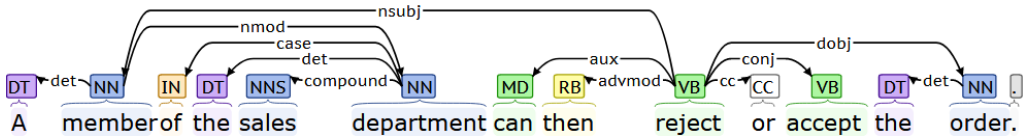
- الشجرة النحوية Parsing Tree. في الواقع يمكن اعتبار الشجرة النحوية الخرج الناتج عن العمليات السابقة مجتمعة. حيث يتم تمثيل الجملة كشجرة. انظر الشكل 5.2 لمثال توضيحي. الاختصارات المكتوبة، مثل NN تعني اسم مفرد singular noun، NNS تعني اسم جمع plural noun. وللإطلاع على هذه القائمة كاملةً انظر [1].



الشكل 5.2: الشجرة النحوية الناتجة للجملة The children ate the cake with a spoon.

- تحليل التّبعية Dependency Parsing. نظراً لقصور الشجرة النحوية عن إعطاء كامل المعلومات حول الجملة المدروسة، تم اقتراح تمثيل المعلومات النحوية على شكل بيان التبعية Dependency Graph. يوضح هذا البيان العلاقات النحوية بين كلمات الجملة. إذ يتكون من عقد تمثّل الكلمات ووصلات تحدد العلاقة بين هذه الكلمات. يبيّن الشكل 6.2 مثالاً على خرج تحليل التبعية. الاختصارات المكتوبة للعقد هي ذاتها المستخدمة في حالة الشجرة النحوية. الاختصارات المكتوبة للوصلات، مثل det تربط بين الاسم وأداة التعريف المرتبطة به، nsubj تربط بين الفعل والفاعل. وللإطلاع على هذه القائمة كاملةً انظر [7].

- حل الغموض في حالات الإحالة Anaphora Resolution. الإحالة Anaphora هي استخدام



الشكل 6.2: بيان التبعية الناتج للجملة A member of the sales department can then reject or accept the order.

ضمائر أو تعابير للإشارة إلى اسم أو تعبير تم ذكره في سياق سابق في النص. فإذا تأملنا المثال التالي:

They buy the issue, then resell it to the public.

إن it تشير إلى issue. وفي هذه الحالة يكون المشار إليه واقعاً في نفس الجملة. أمّا في حالة They فيكون المشار إليه واقعاً في جملة سابقة.

3.2 أهم الأبحاث المشابهة

كما شرحنا في الفقرة 1.2 وتحديداً الفقرة 2.1.2، إن أولى الخطوات اللازمة لتنجيز أي نظام يستخدم تعلم الآلة هي تحديد السمات التي سيتم استخدامها. فهذا المشروع يتعامل مع النصوص وكون المسألة المطروحة هي تقييم جودة هذه النصوص من ناحية سهولة القراءة، فمن المهم جداً معرفة السمات التي ستعبّر عن وتوصّف جودة نص.

بالعودة إلى العديد من الأوراق العلمية التي تتقاطع مع هذا المشروع، تم تجميع عدد من الأفكار والمنهجيات التي تم تبنيها والاعتماد عليها كإطار عمل ضمن المشروع. سنذكر في هذه الفقرة أهم الأوراق العلمية التي تمت دراستها، وأهم الأفكار والمنهجيات المتبعة فيها. بالإضافة إلى السمات وخوارزميات تعلم الآلة المستخدمة لحل مسائل مشابهة للمسألة المطروحة في هذا المشروع.

إن فكرة تقييم النصوص من ناحية صعوبة القراءة بشكل موضوعي (غير شخصي) بدأت تقريباً منذ قرن. الأفكار الأولية التي واجهت هذا الموضوع هي عبارة عن علاقات رياضية بسيطة. ولقد اعتَمَدَت على خصائص سطحية في النص المدروس. مثل متوسط طول الجملة، ومتوسط طول الكلمة. فمثلاً إن Flesch Score يعطى بالعلاقة

$$\text{Flesch Score} = 206.835 - 1.015 \cdot \frac{\#\text{words}}{\#\text{sentences}} - 84.6 \cdot \frac{\#\text{syllables}}{\#\text{words}}$$

أي يمثل علاقة خطية لمتوسط طول الجملة بالكلمات، ومتوسط طول الكلمة بالمقاطع الصوتية. وكلما كان

هذا المقدار أكبر، كلما كان النص أسهل للقراءة. في [5] أُجريب مقارنة بين عدد واسع من هذه العلاقات. وفي [6] تم رسم ودراسة توزع عدد من هذه المعايير على عدد كبير من النصوص. على الرغم من كون هذه العلاقات تبدو سطحية من ناحية التمثيل اللغوي للنص، تم اعتمادها بشكل واسع لمدة من الزمن.

كما ظهرت نماذج أكثر تعقيداً تعتمد على مفهوم الـ n-gram. وهو نموذج إحصائي لتمثيل اللغة؛ يعبر عن احتمال ورود كلمة ضمن سياق معيّن، أو احتمال ورود جملة أو سياق معيّن. وهو تمثيل بسيط للانتروبية في اللغة الانكليزية. استخدم هذا المفهوم بالإضافة إلى سمات أخرى بُنيت فوقه بالاستفادة من مفهوم الانتروبية في عدد من الدراسات أهمها [9,3]. وكانت النتائج أفضل بشكل واضح عن نتائج المعايير السابقة والمعتمدة على علاقات رياضية بسيطة. حيث تم الاعتماد على خوارزميات تعلم الآلة. الخوارزمية الأكثر استخداماً والتي حققت أفضل النتائج كانت الـ SVM.

مع تطور الأدوات في معالجة اللغات الطبيعية، بات من الممكن أن نأخذ بعين الاعتبار مؤشرات أقوى، مفرداتية ونحوية وغيرها. في [11,10] تمت دراسة ومقارنة العديد من السمات التي تعتمد بشكل أساسي على تقدم الأدوات في معالجة اللغات الطبيعية. حيث تمت مقارنة عدّة سمات، وبعدّة أنواع. مثل التنوع في الأفعال، والكلمات المستخدمة، وكثافة الأسماء المستخدمة في النص. بالإضافة إلى الترابط بين الجمل باستخدام بيان التبعية والعديد غيرها. أفضل نتيجة تم تحقيقها كانت باستخدام الـ SVM على مجموعة جزئية من السمات المدروسة.

كما تمت دراسة تعقيد النصوص المكتوبة من قبل الطلاب الذين يتعلمون اللغة الانكليزية. تم ذلك تحت ما يسمى أبحاث تعلم اللغات second language acquisition. حيث تمت دراسة عدد من المؤشرات التي تعبر عن تعقيد النص، بما يفيد في دراسة تحسن الطلاب أثناء تعلمهم للغة. كما أن دراسات لاحقة قامت بأتمتت آليات حساب هذه المؤشرات أهمها [12]. فكان تغيير هذه المؤشرات مع الزمن، يبيّن مستوى تحسن الطلاب في تعلم اللغة. وكان أول استخدام لهذه الدراسات لبناء نموذج تعلم آلة لتقييم جودة النصوص هو في [15]. حيث تم الاعتماد على سمات مستوحاة بشكل مباشر من هذه المؤشرات. معظم هذه السمات تعبر عن تعقيد تراكيب الجمل. مثل وجود جمل شرطية أو جمل معطوفة على بعضها وهكذا.

استخدمت بعض الأبحاث سمات تعبر عن نسبة ورود كلمات معينة ضمن النص. فمثلاً إن ورود كلمات متقدمة ضمن النص وبتواتر عالي قد يكون مؤشر جيد على كون النص بمستوى متقدم. هذه الكلمات تكون غالباً منتقاة من مصدر تعليمي. فقد تم استخدام كلمات من المصدر The Academic Word List[1] في [26,18,15]. و تم استخدام كلمات من المصدر English Vocabulary Profile[2] في [23].

[1] للإطلاع على هذه القائمة كاملةً انظر [2].

[2] لمزيد من التفاصيل انظر http://www.englishprofile.org.

حيث تبيّن أن لهذه السمات دور جيد في تحسين أداء النماذج الناتجة.

سنوضح لاحقاً وبتفصيل أكبر في الفقرة 4.3 السمات المستخدمة في هذا المشروع. يجب أيضاً التنويه إلى المعطيات المستخدمة في هذه الدراسات. إن معظم المعطيات هي من مصدر تعليمي، مثل مجلات تعليمية للأطفال حيث أن المقالات مصنفة بحسب الفئة العمرية المناسبة. سنتحدث لاحقاً في الفقرة 3.3 عن مجموعة المعطيات المستخدمة في هذا المشروع وخصائصها.

الفصل الثالث

تصميم النظام

يبيّن هذا الفصل منهجية العمل المتبعة خلال تنفيذ المشروع. ويشرح النظام على مستوى عالي من التجريد وفق مخططات صندوقية. كما يسرد السمات التي استخراجها من النصوص. ويسرد خوارزميات تعلم الآلة التي تم استخدامها.

1.3 منهجية العمل

نتعامل مع المسألة المطروحة ضمن المشروع على أنها مسألة تصنيف مقروئية نص مكتوب باللغة الانكليزية وفق عدّة مستويات. فالغاية المرجوّة هي معرفة مستوى صعوبة نص معيّن وإلى أي مستوى ينتمي. سؤال قد يتم طرحه هنا وهو: ما هو عدد المستويات وما هو التفاوت ومعيار المقارنة بينها؟ الإجابة هي أن عدد المستويات والفروقات بينها يتم تحديده ضمن المعطيات التي ستُستخدم لتدريب المصّنف. فقد تكون هذه المعطيات مفصولة إلى أي عدد من المستويات. ولكن وجِبَ أن يكون معيار المقارنة بين هذه المستويات هو مقروئية النصوص وذلك لكي يحقق التطبيق الناتج الهدف المرجو منه.

تبدأ أنظمة تعلم الآلة عادة بجمع المعطيات. في حالتنا هذه، نريد جميع عدد كبير من النصوص المصنّفة بشكل مسبق وصحيح إلى مستوى صعوبة مقروئيتها. لم نحتاج إلى القيام بهذه المرحلة ضمن المشروع بسبب توافر هكذا معطيات. المرحلة التي تليها هي مرحلة استخراج السمات. إذ يتم التعبير عن المعطيات الخام بأشعة من السمات يمكن لخوارزميات تعلم الآلة إجراء عمليات حسابية عليها. وبعد استخراج السمات، يتم اختيار خورازمية تعلم

الآلة المناسبة وضبط برامتراتها لتدريبها على جزء من هذه المعطيات (مجموعة التدريب) واختبارها على الجزء الآخر (مجموعة الاختبار). وذلك لتقييم مستوى أدائها ومعرفة الجدوى من استخدامها.

أخيراً بعد إجراء عملية التدريب والحصول على مُصنّف جاهز للاستخدام، نقوم لأجل نص جديد باستخراج سماته واستخدام المصنّف للحصول على مستوى مقروئية هذا النص.

2.3 المخططات الصندوقية للنظام

كما رأينا في الفقرة السابقة، توجد عدّة مراحل لتنجيز النظام بشكل كامل. ودون الخوض في كثير من التفاصيل، سنعتبر أنه توجد مرحلتان أساسيتان لتنجيز المشروع. الأولى هي للحصول على مصنّف جاهز للاستخدام. الثانية هي استخدام هذا المصنّف.

يبيّن الشكل 1.3 المخطط الصندوقي للنظام الذي تم تنجيزه للحصول على المصنف وتقييم أدائه. بينما يبيّن الشكل 2.3 المخطط الصندوقي لاستخدام هذا المصنف.

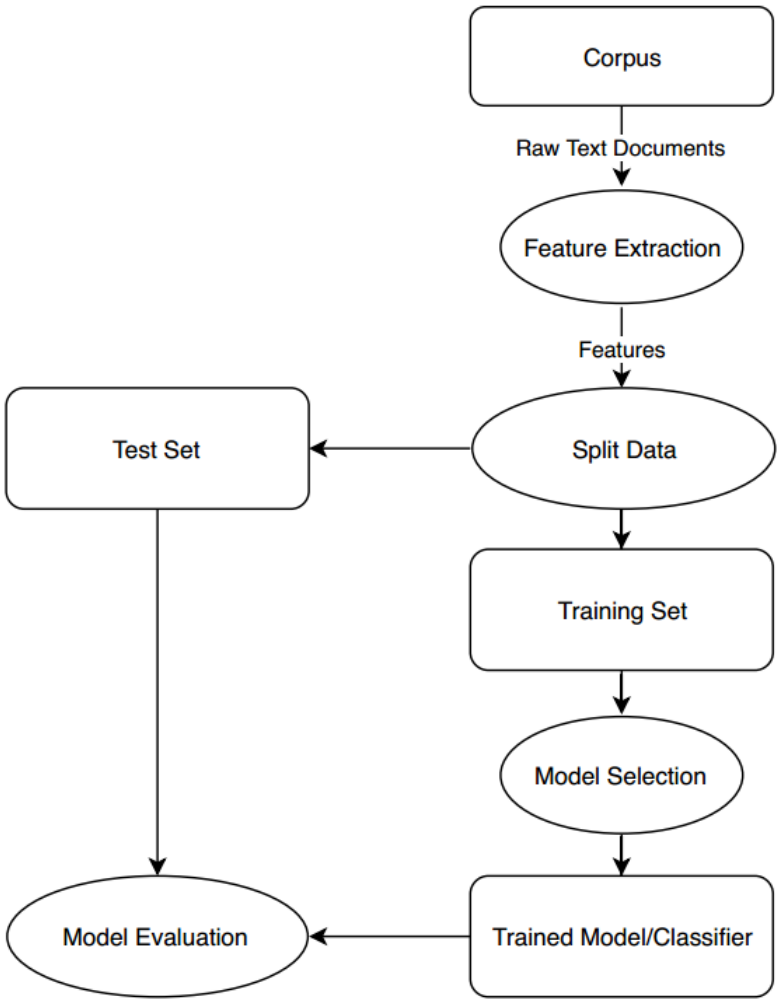
3.3 المعطيات المستخدمة

تبّين هذه الفقرة المعطيات المستخدمة ضمن المشروع وخصائصها.

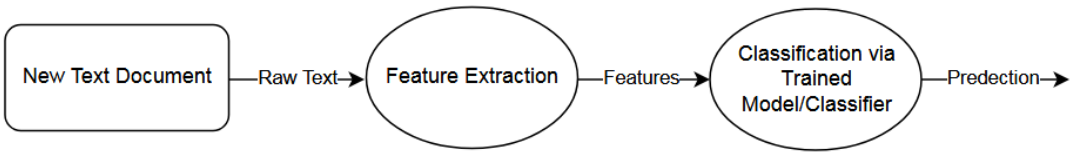
1.3.3 One Stop English Corpus (OSE)

يعود الفضل في تجميع هذه المعطيات إلى [26]. تم تجميع هذه المعطيات من الموقع http://www.onestopenglish.com في الفترة ما بين 2013 – 2016. وهو موقع تعليمي بأكثر من 700,000 مستخدم من 100 دولة.

أحد سمات هذا الموقع هو وجود درس تعليمي أسبوعي له طابع إخباري يحوي مقالات من الصحيفة البريطانية The Guardian. إذ تتم إعادة صياغة مقالات هذه الصحيفة من قبل المدرسين لتناسب ثلاثة مستويات من الطلاب (مبتدئ elementary، متوسط intermediate، متقدم advanced). أي أنه تتم إعادة صياغة



الشكل 1.3: المخطط الصندوقي للحصول على المصنّف وتقييم أدائه.



الشكل 2.3: المخطط الصندوقي لاستخدام المصنّف.

محتوى الصحيفة الأصلي إلى ثلاث نسخ متدرجة الصعوبة من حيث مقروئيتها مع المحافظة على أكبر قدر من فحوى المحتوى الأصلي. يبيّن الجدول 1.3 عيّنة من هذه المعطيات.

تُبيّن لنا طريقة جمع هذه المعطيات أهميتها بالنسبة للمشروع. إذ إن معيار المقارنة بين هذه المستويات هو مقروئية النصوص من ناحية تعقيد تراكيب الجمل أو بساطتها وذلك لنصوص لها نفس الفحوى. وإجراء الاختبارات عليها سيوضح الجدوى من استخدام هذا النظام في تحليل مقروئية النصوص من ناحية الصياغة.

Reading Level	**Sample Text**
Elementary	To tourists, Amsterdam still seems very liberal. Recently the city's Mayor told them that the coffee shops that sell marijuana would stay open, although there is a new national law to stop drug tourism. But the Dutch capital has a plan to send antisocial neighbours to scum villages made from shipping containers, and so maybe now people wont think it is a liberal city any more.
Intermediate	To tourists, Amsterdam still seems very liberal. Recently the city's Mayor assured them that the city's marijuana-selling coffee shops would stay open despite a new national law to prevent drug tourism. But the Dutch capitals plans to send nuisance neighbours to scum villages made from shipping containers may damage its reputation for tolerance.
Advanced	Amsterdam still looks liberal to tourists, who were recently assured by the Labour Mayor that the city's marijuana-selling coffee shops would stay open despite a new national law tackling drug tourism. But the Dutch capital may lose its reputation for tolerance over plans to dispatch nuisance neighbours to scum villages made from shipping containers.

جدول 1.3: عيّنة من جمل الـ OSE المصنفة إلى ثلاثة مستويات.

تتألف هذه المعطيات من 567 نص موزعين بالتساوي إلى المستويات الثلاثة، أي يوجد 189 نص في كل مستوى. ويبّن الجدول 2.3 بعض الإحصائيات الوصفية لنصوص هذه المعطيات. وهي متوسط طول النص، والانحراف المعياري لطول النص وذلك للمستويات الثلاثة كلاً على حدا. وإن الواحدة المستخدمة لطول النص هي الكلمة. نلاحظ (كما هو متوقع) أن الطول الوسطي للنصوص يتزايد مع تزايد المستوى. وأن الانحراف المعياري لطول النصوص كبير مما يجعل طول النص معيار غير كافي لتحديد صعوبته.

هذه النصوص متاحة على الرابط https://github.com/nishkalavallabhi/OneStopEnglishCorpus ويجب التنويه إلى أن هذه المعطيات لم تسخدم كما هي، بل تم إجراء تنضيف شبه يدوي عليها. حيث أنه

Reading Level	Avg. Num. Words	Std. Dev.
Elementary	533.17	103.79
Intermediate	676.59	117.15
Advanced	820.49	162.52

جدول 2.3: إحصائيات وصفية لنصوص الـ OSE.

وُجِدَت مجموعة من المحارف الغريبة التي تم استبدالها بمحارف مناسبة بحسب سياق ورودها ضمن النصوص. فقد سبب بعض هذه المحارف مشاكل في قراءة النص أو استخدام مكتبات معالجة اللغات الطبيعية. بالإضافة إلى كونها تشكل تشويش في المعطيات. فيمكن اعتبار أن أحد منجزات هذا المشروع هو تنظيف معطيات الـ OSE بالكامل وبإشراف شبه يدوي.

4.3 السمات المستخدمة

تم استخدام مجموعة جزئية من السمات التي تم استخدامها في الأبحاث السابقة. وقد قمنا بتصنيف السمات المستخدمة وفق عدّة فئات. وذلك بحسب نوع المعلومات التي تحملها من النص.

1.4.3 السمات التقليدية Traditional Features

وهي سمات سطحية تحمل معلومات عامة عن النص.

1. عدد المحارف ضمن النص CharacterCount.
2. عدد الكلمات ضمن النص WordCount.
3. متوسط طول الكلمة AverageWordLength.
4. عدد الجمل ضمن النص SentenceCount.
5. متوسط طول الجملة بالكلمات AvgSentenceLengthInWords.
6. متوسط طول الجملة بالمحارف AvgSentenceLengthInChars.

2.4.3 السمات المفرداتية Lexical Features

وهي السمات التي تحمل الصفات المفرداتية للنص؛ مثل تنوع المفردات المستخدمة فيه من أفعال وأسماء وغيرها.

7. عدد الكلمات المتمايزة (المختلفة) ضمن النص UniqueWordsCount.
8. نسبة عدد الكمات المختلفة إلى عدد الكلمات الكلي Type-Token Ratio (TTR). وتساوي T/N حيث T عدد الكلمات المختلفة types، وN عدد الكلمات الكلي tokens. إذ بملاحظة كون طول النص يلعب دور أساسي في البند السابق بالإضافة إلى معظم السمات التقليدية، فهذه السمة تحاول تهميش دور طول النص.
9. ولتصحيح المعامل السابق بشكل أنسب؛ أي لاستبعاد دور طول النص. تم اقتراح النسبة $T/\sqrt{2N}$ وتسمى Corrected Type-Token Ratio.
10. تنوع الأسماء ضمن النص NounVariation. وهو نسبة عدد الأسماء إلى عدد الكلمات المفرداتية lexical tokens. إذ يقصد بالكلمات المفرداتية كل من الأسماء nouns، والأفعال verbs، والصفات adjectives، والأحوال adverbs.
11. تنوع الأفعال ضمن النص VerbVariation. وهو نسبة عدد الأفعال إلى عدد الكلمات المفرداتية.
12. تنوع الأفعال ضمن النص AdjectiveVariation. وهو نسبة عدد الصفات إلى عدد الكلمات المفرداتية.
13. تنوع الأفعال ضمن النص AdverbVariation. وهو نسبة عدد الأحوال إلى عدد الكلمات المفرداتية.
14. نسبة عدد الأفعال المتمايزة وبعد تجذيرها إلى عدد الأفعال الكلي VerbVariation1. ويساوي T_v/N_v حيث N_v هو عدد الأفعال الكلي، وT_v هو عدد الأفعال المتمايزة (المختلفة) بعد إجراء عملية التجذير.
15. النسبة T_v^2/N_v وتسمى SquaredVerbVariation1.
16. النسبة $T_v/\sqrt{2N_v}$ وتسمى CorrectedVerbVariation1.
17. الكثافة المفرداتية LexicalDensity. وهي نسبة عدد الكلمات المفرداتية إلى عدد الكلمات الكلي.

3.4.3 السمات النحوية Syntactic Features

وهي السمات التي تعبّر عن الصياغة والتراكيب المستخدمة ضمن النص. لقد تم تعريف عدد من التراكيب النحوية، ثمّ استخدامها لتعريف سمات نحوية. يعود الفضل في تعريف هذه التراكيب وتعريف آلية حسابها ودراستها إلى [12]. سنكتفي هنا بسرد أسماء هذه التراكيب. وذلك لكون تعريفها معقّد وقد يحتاج فهم بعضها لخبير في اللغة الانكليزية. وللتفاصيل الدقيقة حول تعريفها وطريقة حسابها يمكن العودة إلى [12]. هذه التراكيب هي:

Sentences (S), Clauses (C), Dependent Clauses (DC), T-units (T), Complex T-units (CT), Coordinate Phrases (CP), Complex Nominals (CN).

وإن السمات النحوية المستخدمة هي:

18. متوسط ارتفاع الشجرة النحوية لكل جملة من النص AvgTreeDepth.
19. متوسط طول C بالكلمات.
20. متوسط طول T بالكلمات.
21. النسبة C/S.
22. النسبة C/T.
23. النسبة CT/T.
24. النسبة DC/C.
25. النسبة DC/T.
26. النسبة CP/C.
27. النسبة CP/T.
28. النسبة T/S.
29. النسبة CN/C.
30. النسبة CN/T.

5.3 الخوارزميات المستخدمة

تم استخدام عدّة خوارزميات وضبط برامتراتها وفق عدّة قيم. أهم الخوارزميات التي تم تجريبها هي:

- Naive Bayes: وهي خوارزمية رياضية بسيطة تعتمد على الاحتمالات الشرطية ونظرية الاحتمالات.
- Logistic Regression: من أبسط خوارزميات تعلم الآلة. تستخدم نموذج خطي. وتتميز بأن الخرج الناتج يمثّل احتمال؛ أي أن الخرج لا يكون رقم الصف الذي تنتمي إليه العيّنة المدروسة، وإنما مجموعة قيم تمثّل احتمال إنتماء العينة إلى الصف الأول، واحتمال إنتماء العينة إلى الصف الثاني، وهكذا.
- الشبكات العصبونية Neural Networks: تم تجريبها وفق عدة بُنى وعدة أشكال. أدائها كان سيِّئ وقد احتاجت وقت طويل أثناء عملية التدريب والاحتبار.
- J48: تعتمد على بناء شجرة قرار للتصنيف. وتستخدم مفهوم الانتروبية. وقد أعطت نتائج سيئة.
- SVM: أعطت أفضل النتائج. وذلك باستخدام النواة الخطية، وضبط المعامل C إلى 1. للتفاصيل عُد إلى الفقرة 3.1.2.

الفصل الرابع

التصميم البرمجي والتنجيز

نبيّن في هذا الفصل التصميم البرمجي للنظام وطريقة تنجيزه والأدوات المستخدمة لذلك. ونقوم بشرح مخططات الصفوف للحزم البرمجية. وسرد القرارات التصميمة المعتبرة والأنماط التصميمية Design Patterns المستخدمة.

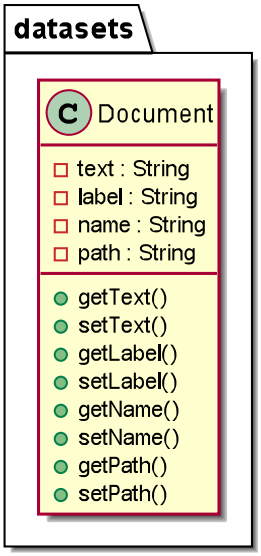
1.4 الأدوات المستخدمة

أهم الأدوات التي تم استخدامها أثناء تنجيز المشروع هي:

- بيئة التطوير المستخدمة هي IntelliJ IDEA. وهي بيئة لتطوير برامج الجافا. توفر عدد كبير من السمات. يمكن تحميلها عبر الرابط https://www.jetbrains.com/idea/download.
- تم استخدام البرنامج Git كـ version control system. وإن الكود البرمجي المكتوب مرفوع على الموقع GitHub على الرابط https://github.com/fresher96/text-readability-assessment.
- تم استخدام مكتبة ستانفورد Stanford Parser لإجراء عمليات معالجة اللغات الطبيعية. المزيد في الفقرة 5.3.4.
- تم استخدام الأداة Weka لتطبيق خوارزميات تعلم الآلة. المزيد في الفقرة 4.4.

2.4 قراءة المعطيات

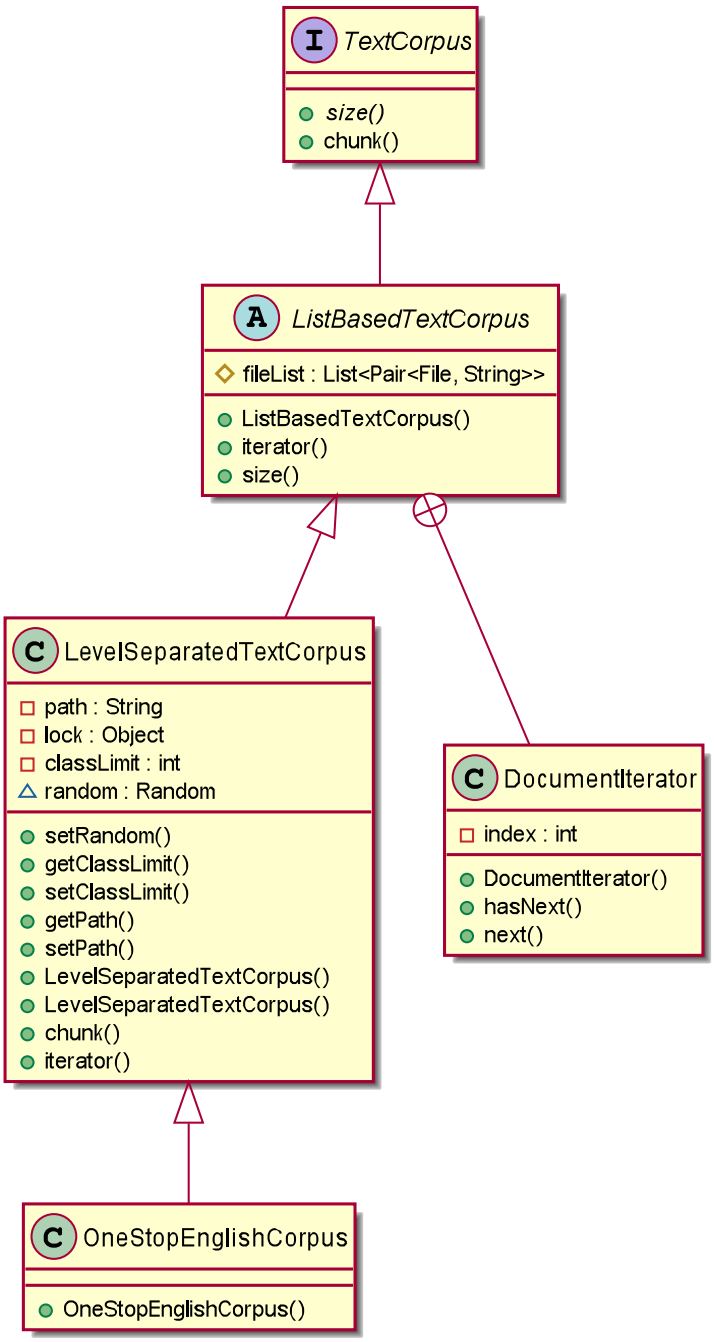
تم بناء الحزمة datasets للتعامل مع المعطيات. أي لقراءة النصوص وكتابة السمات. تحوي هذه الحزمة صف وحيد Document وهو الصف الأساسي المستخدم ليحمل معلومات النص مثل اسمه ومساره وغيرها. يبيّن الشكل 1.4 مخطط الصفوف لهذه الحزمة. كما تحوي هذه الحزمة حزمتين جزئيتين هما الحزمة corpora والحزمة writers.



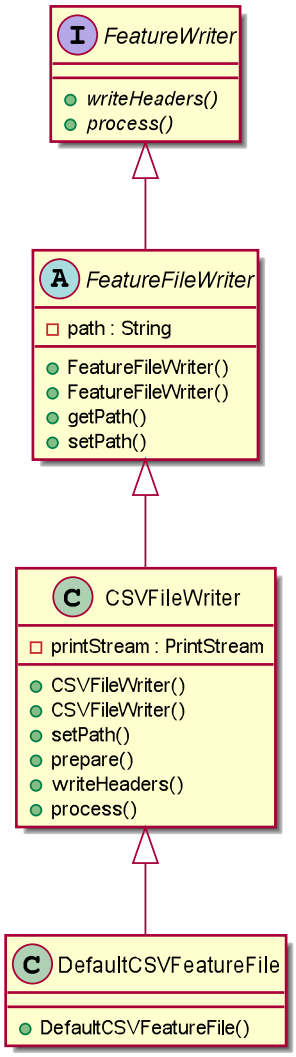
الشكل 1.4: مخطط الصفوف للحزمة datasets.

الحزمة corpora فيها مجموعة من الصفوف المستخدمة لقراءة مجموعة كبيرة من النصوص والمرور عليها ومعالجتها. إذ أن الصفوف خارج هذه الحزمة تستخدم الواجهة TextCorpus. ويمكن توسيع هذه الحزمة بإنشاء صف جديد ينجّز هذه الواجهة. حيث يجب أن يعرّف آلية الحصول على النصوص المكتوبة وتصنيفاتها. تم استخدام النمط Iterator design pattern لتحقيق ذلك. إذ وجدناه مناسباً ويقوم بتأدية الغرض اللازم. يبيّن الشكل 2.4 مخطط الصفوف لهذه الحزمة.

وتم بناء الحزمة writers لكتابة ملف فيه السمات التي تم استخراجها من هذه النصوص. يبيّن الشكل 3.4 مخطط الصفوف لهذه الحزمة. الواجهة الأساسية التي يتم استخدامها خارج هذه الحزمة هي FeatureWriter. الصف المكتوب والذي ينجزها يقوم بكتابة السمات على ملف بلاحقة CSV (Comma Separated Values). يمكن بإضافة صف ينجز هذه الواجهة تعريف آلية لكتابة السمات بأي صيغة أخرى بحسب المطلوب كـ XML مثلاً.



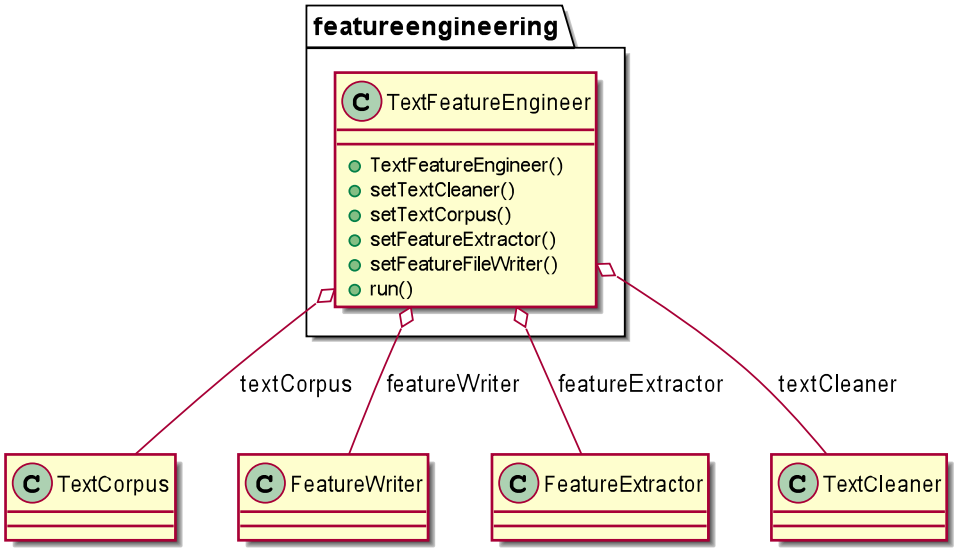
الشكل 2.4: مخطط الصفوف للحزمة datasets.corpora.



الشكل 3.4: مخطط الصفوف للحزمة datasets.writers.

3.4 استخراج السمات

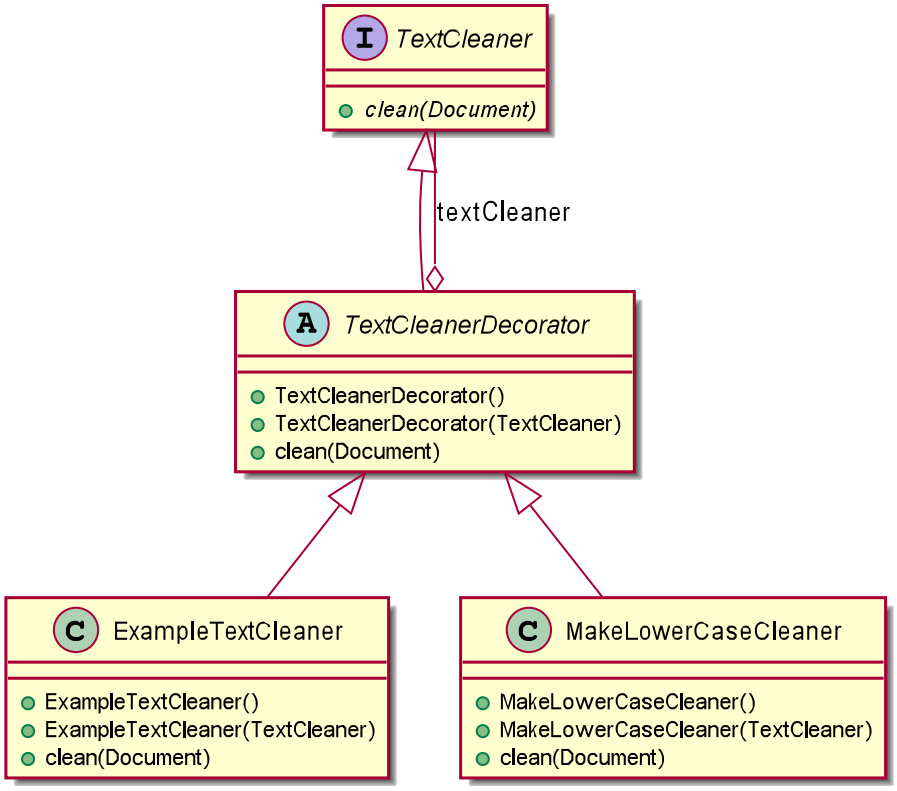
تم بناء الحزمة featureengineering لتحوي الصفوف المسؤولة عن استخراج السمات من النصوص. تحوي هذه الحزمة صف واحد وأربع حزم جزئية. الصف الموجود TextFeatureEngineer هو صلة وصل، إذ يقوم باستخدام الصف اللازم لقراءة النصوص واستخراج السمات منها ثمّ كتابة ملف السمات. وأثناء عمله يقوم بطباعة معلومات مفيدة. مثل اسم ورقم الملف الذي تتم معاجته حالياً، والوقت المستغرَق للمعالجة. وبعد الانتهاء يذكر عدد الملفات الذي حدث خطأ أثناء معالجتها. ننوه إلى أن تنفيذ عملية استخراج السمات تستهلك وقت يتراوح بين ساعة وساعتين. والسبب الأساسي في استهلاك هذا الوقت الكبير هو استخدام مكتبات معالجة اللغات الطبيعية. يوضح الشكل 4.4 مخطط الصفوف لهذه الحزمة.



الشكل 4.4: مخطط الصفوف للحزمة featureengineering.

1.3.4 الحزمة cleaners

تحوي هذه الحزمة على الصفوف التي تقوم بتنظيف النص بشكل آلي قبل البدء بعملية استخراج السمات. مثل أن يتم تحويل جميع الأحرف إلى حروف صغيرة، أو حذف علامات الترقيم، إلخ. تم استخدام النمط Decorator design pattern. وذلك للسماح باستخدام عدّة صفوف تقوم بالتنظيف ودون تحديد عددها وبشكل سهل الاستخدام. يبيّن الشكل 5.4 مخطط الصفوف لهذه الحزمة.



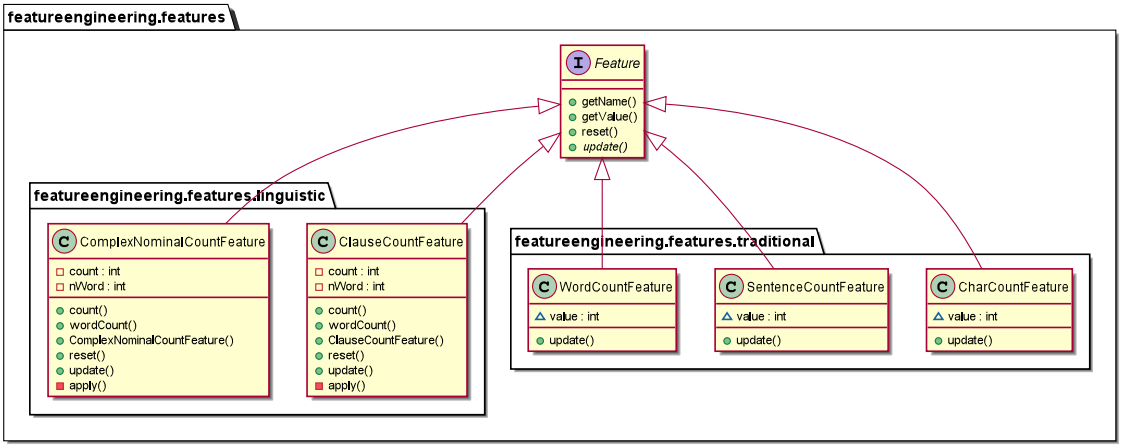
الشكل 5.4: مخطط الصفوف للحزمة featureengineering.cleaners.

2.3.4 الحزمة features

تحوي هذه الحزمة على السمات التي تم تنجيزها. كل صف يمثل سمة. وجميع هذه الصفوف تنجّز الواجهة Feature. ويمكن بإنشاء صفوف جديدة تُنجّز هذه الواجهة إضافة سمات جديدة وتوسيع الحزمة. يبيّن الشكل 6.4 جزء من مخطط الصفوف لهذه الحزمة. نلاحظ أنه تم تقسيم السمات بحسب طبيعتها إلى عدّة حزم جزئية.

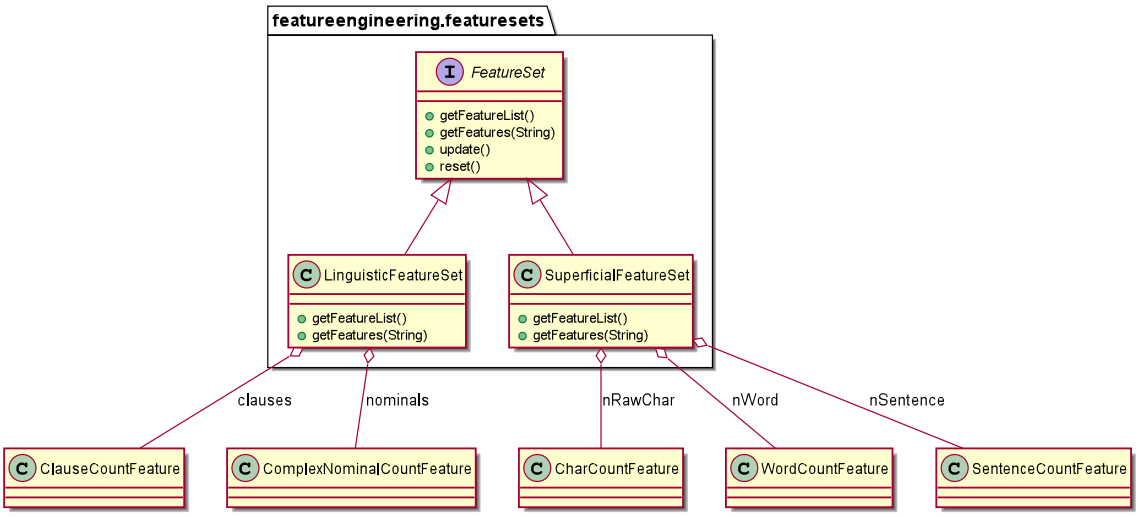
3.3.4 الحزمة featuresets

كل صف من صفوف هذه الحزمة يمثّل مجموعة سمات مترابطة. يبيّن الشكل 7.4 عيّنة جزئية من مخطط الصفوف لهذه الحزمة. الهدف من هذه الصفوف هو تسهيل استخدام السمات التي تم تنجيزها ضمن سياق آخر. يمكن للمستخدم (المستخدم في هذه الحالة هو مبرمج) بإنشاء صف يُنجّز الواجهة FeatureSet واستخدام



الشكل 6.4: عيّنة من مخطط الصفوف للحزمة featureengineering.features.

السمات التي يريدها من الحزمة features. أيضاً وجود هذه الصفوف يسمح بتركيب عدد من السمات؛ فمثلاً يمكن استخدام الصف WordCountFeature لحساب عدد الكلمات ضمن النص، واستخدام الصف SentenceCountFeature لحساب عدد الجمل، ثمّ حساب متوسط طول الجملة بتقسيم عدد الكلمات على عدد الجمل. تم استخدام هذا التصميم لفصل السمات عن بعضها بحيث تكون مستقلة ويمكن استخدام كل منها على حدا، وأيضاً لتفادي إعادة الحسابات ورفع الكفاءة.



الشكل 7.4: عيّنة من مخطط الصفوف للحزمة featureengineering.featuresets.

4.3.4 الحزمة extractors

يبيّن الشكل 8.4 مخطط الصفوف لهذه الحزمة. المكون الأساسي فيها هو الواجهة FeatureExtractor. إذ يعتبر المكوّن الأساسي في عملية استخراج السمات. له التابعين ()getFeatureList الذي يعيد أسماء السمات. والتابع (String)extract الذي يعيد قيمة السمات التي تم استخراجها للنص. يمكن تنجيز هذه الواجهة بعدّة طرق.

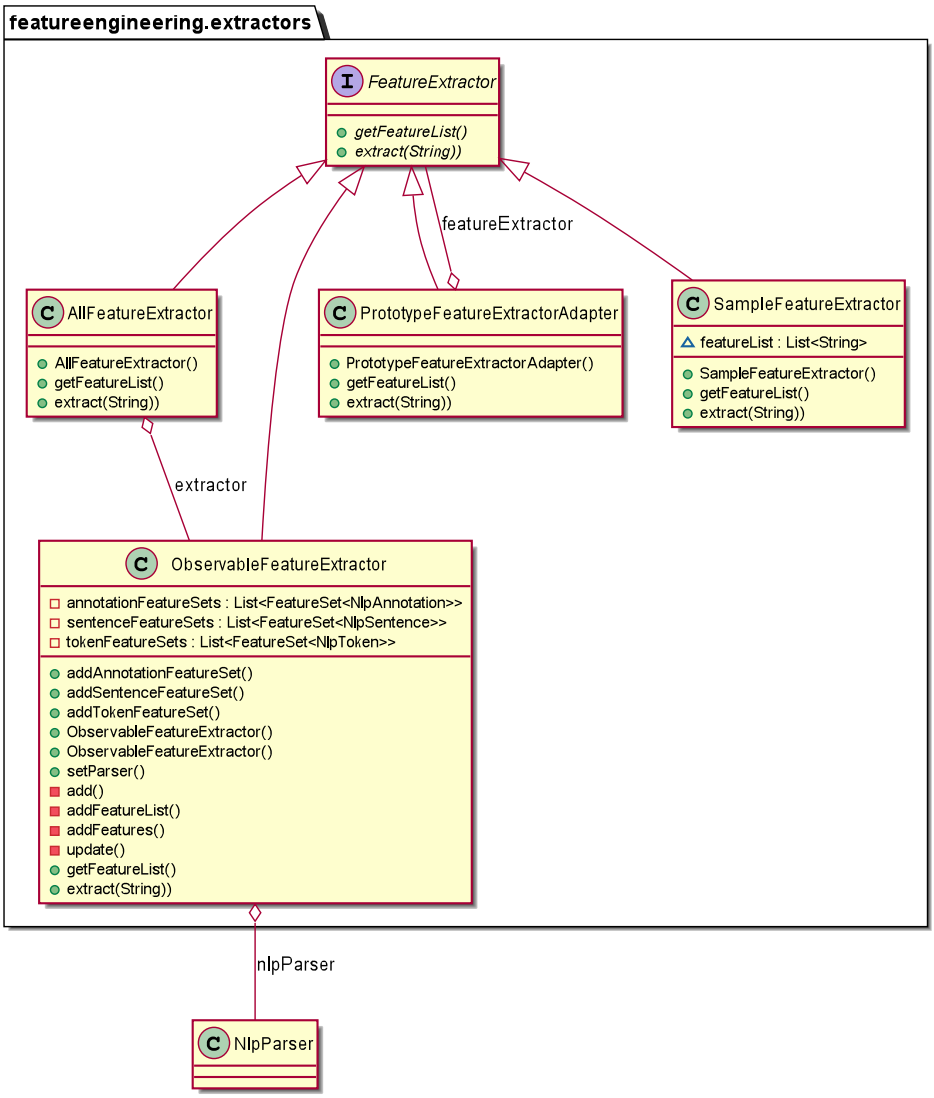
الصف الذي تم إنشائه لتنجيز هذه الواجهة يستخدم النمط Observer design pattern. وهو الصف ObservableFeatureExtractor. يحوي هذا الصف على مجموعة من الـ FeatureSet كل منها هو Observer. تم استخدام هذا النمط لكون عملية إعراب النص parsing باستخدام مكتبة معالجة اللغات الطبيعية يستهلك وقت (حوالي 6 ثواني للنص الواحد). فبعد عملية الإعراب يقوم الصف بتنبيه مجموعات السمات هذه والتي تقوم بدورها بتنبيه السمات فتتحدث قيمها بحسب التغيير الجديد.

5.3.4 الحزمة nlp

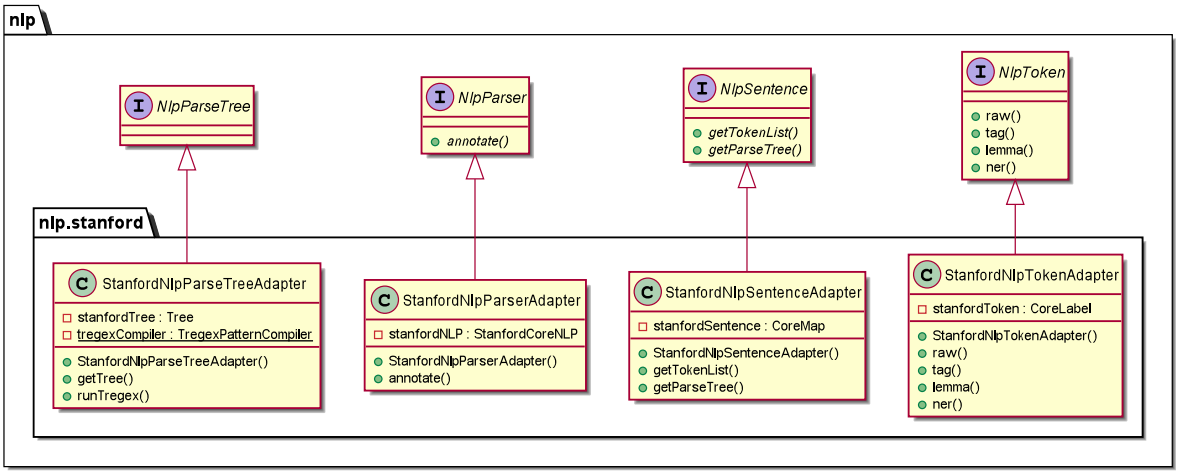
تم بناء هذه الحزمة لتغليف وتوحيد الواجهة البرمجية API لمكتبات معالجة اللغات الطبيعية. فلقد استخدمنا في هذا المشروع مكتبة ستانفورد Stanford Parser[1] لذلك. فهي مكتوبة بلغة جافا. سهلة الاستخدام. تقدم جميع الحسابات المطلوبة، ولكنها تستغرق وقت كبير. وإن تصميم الحزمة nlp بهذا الشكل يسمح بتغيير المكتبة المستخدمة لمعالجة اللغات الطبيعية دون أي تغيير على باقي المكونات البرمجية. كما تتميز مكتبة ستانفورد بوجود إضافة إليها تحت اسم Tregex تسمح بكتابة استعلامات وتنفيذها على الشجرة النحوية الناتجة.

تم استخدام النمط Adapter design pattern لتحقيق ذلك. إذ تم تعريف الواجهات الأساسية والتوابع الأساسية. ولاستخدام مكتبة محددة نقوم بتنجيز الواجهات السابقة بحسب الواجهة البرمجة API للمكتبة المستخدمة. فبذلك تم عزل المكتبة المستخدمة عن الكود البرمجي المكتوب. يوضح الشكل 9.4 عيّنة من مخطط الصفوف لهذه الحزمة.

[1] https://nlp.stanford.edu/software/lex-parser.shtml



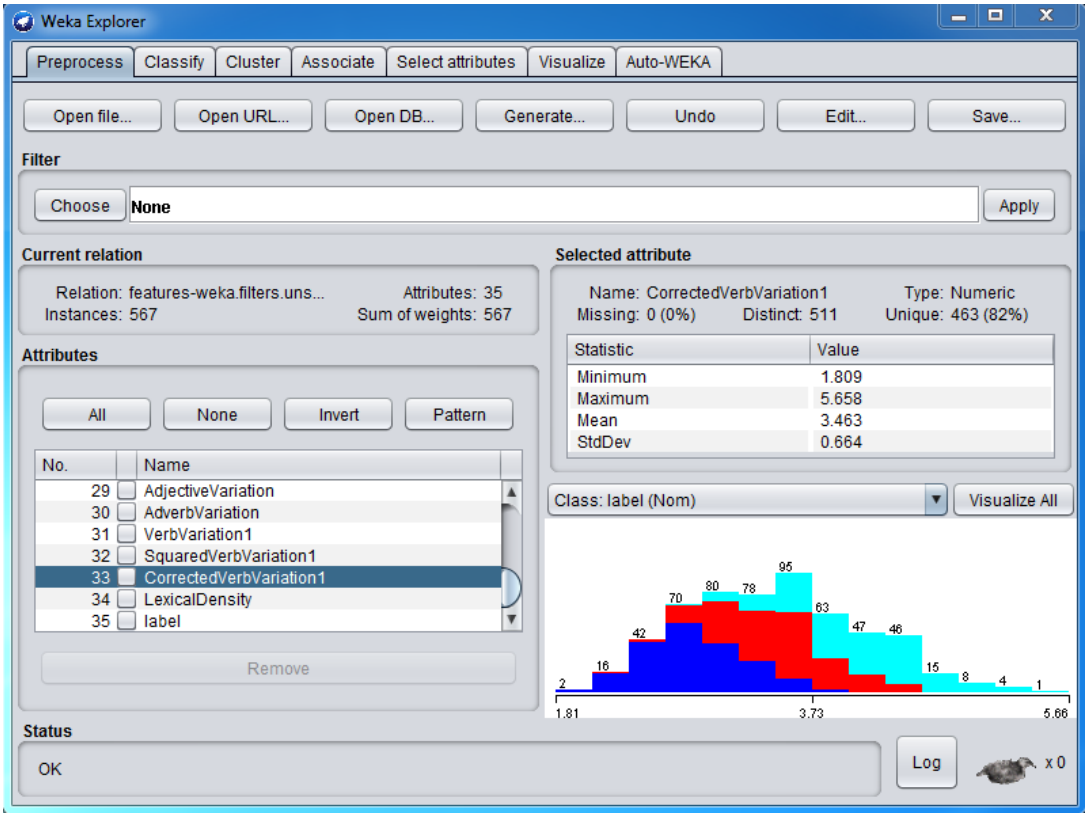
الشكل 8.4: مخطط الصفوف للحزمة featureengineering.extractors.



الشكل 9.4: عيّنة من مخطط الصفوف للحزمة nlp.

4.4 خوارزميات تعلم الآلة

توجد العديد من الأدوات التي تقدم مجموعة واسعة من الوظائف لتطبيق خوارزميات ومفاهيم تعلم الآلة المختلفة. ضمن هذا المشروع، تم اختيار الأداة Weka[2] لذلك. هذه الأداة مكتوبة بلغة البرمجة جافا. توفر واجهة تخاطبية GUI مما يجعل عملية إجراء عدّة تجارب والمقارنة بين مختلف الخوارزميات أمر سهل جداً. يوضح الشكل 10.4 أحد واجهاتها التخاطبية. كما تتميز هذه الأداة بتوفيرها لواجهة برمجية API مما يوفر مرونة كبيرة جداً في استخدامها أو التعديل والتطوير عليها.



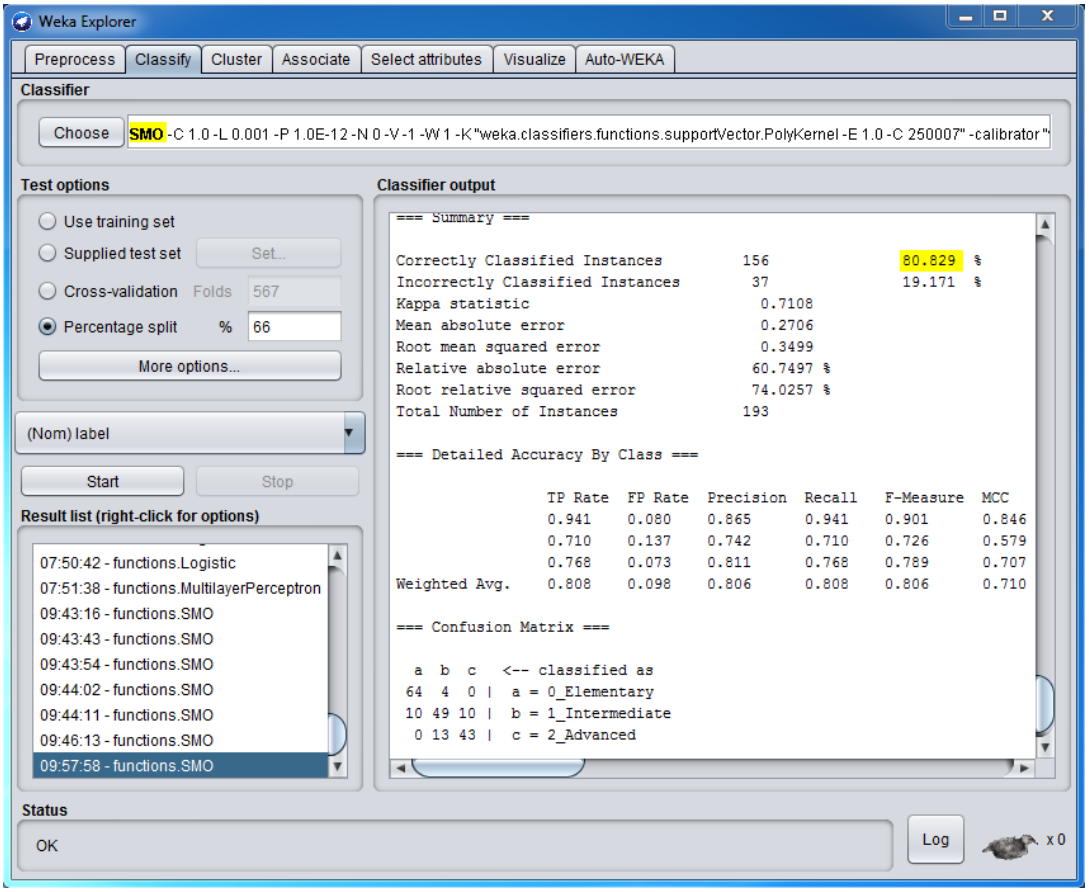
الشكل 10.4: أحد الواجهات التخاطبية للأداة Weka.

كما توفر هذه الأداة عدد واسع من الوظائف المستخدمة بشكل عام ضمن الأنظمة التي تستخدم تعلم الآلة. فتوفر أدوات لعرض المعطيات وبعدّة طرق. وتوفر واجهة لإجراء عدّة اختبارات والمقارنة بين نتائج مجموعة من الخوارزميات المختلفة. فهي تقدم تنجيز جاهز off-the-shelf لمجموعة واسعة جداً من خوارزميات تعلم الآلة. وتسمح بضبط مختلف البارامترات في مختلف هذه الخوارزميات. كما تقوم بعرض وحساب عدد واسع من

[2] https://www.cs.waikato.ac.nz/ml/weka

المعايير والنتائج المهمة المستخدمة لتقييم هذه الخوارزميات. وتوفر آلية لحفظ النماذج الناتجة بعد عملية التدريب لاستخدامها لاحقاً ضمن التطبيق.

كما توفر أدوات مفيدة جداً لعملية ضبط الخوارزميات. تسمح هذه الأدوات بتجريب مجموعة قيم مختلفة للبارمترات الفوقية للخوارزمية واختيار القيم التي تعود بالنتائج الأفضل. بالإضافة إلى أدوات تساعد في مقارنة السمات المستخدمة. ومقارنة أهميتها ومدى مساهمتها في عملية التصنيف. مما يساعد في عملية اختيار مجموعة سمات جزئية Feature Selection في حال كان ذلك ضرورياً لرفع كفاءة النظام. يوضح الشكل 11.4 مثال على تطبيق خوارزمية الـ SVM على ملف السمات التي تم استخراجها من النصوص.



الشكل 11.4: تطبيق خوارزمية الـ SVM باستخدام الأداة Weka.

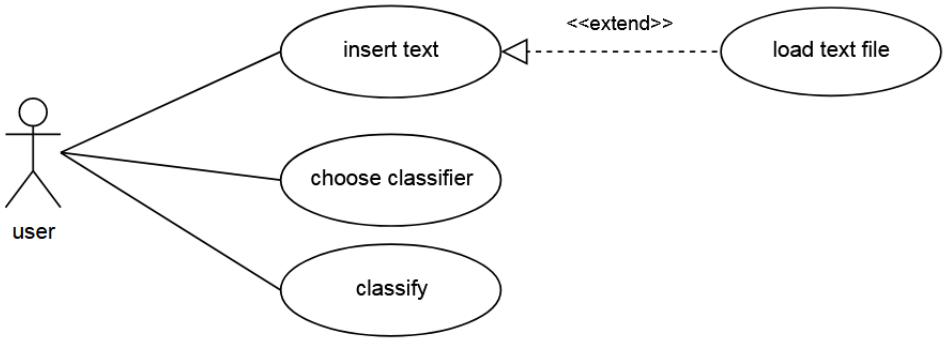
الفصل الخامس

دليل استخدام التطبيق

يبيّن هذا الفصل مخطط حالات الاستخدام للتطبيق النهائي. ويوضّح دليل استخدامه لتصنيف نصوص جديدة.

1.5 حالات الاستخدام

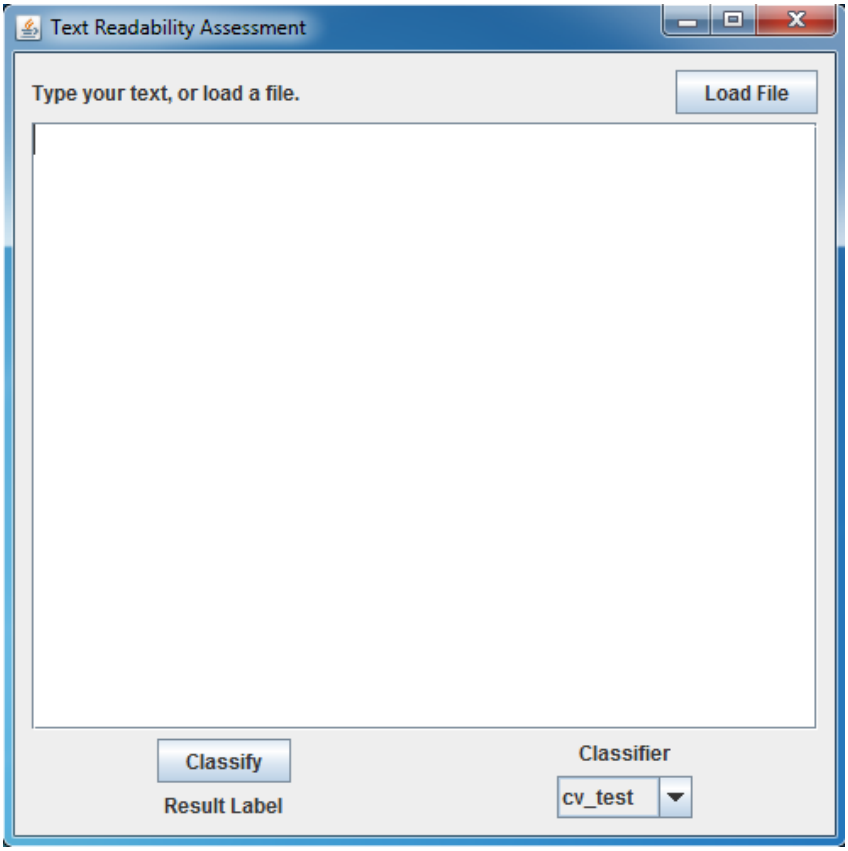
يبيّن الشكل 1.5 مخطط حالات الاستخدام للتطبيق.



الشكل 1.5: مخطط حالات الاستخدام للتطبيق.

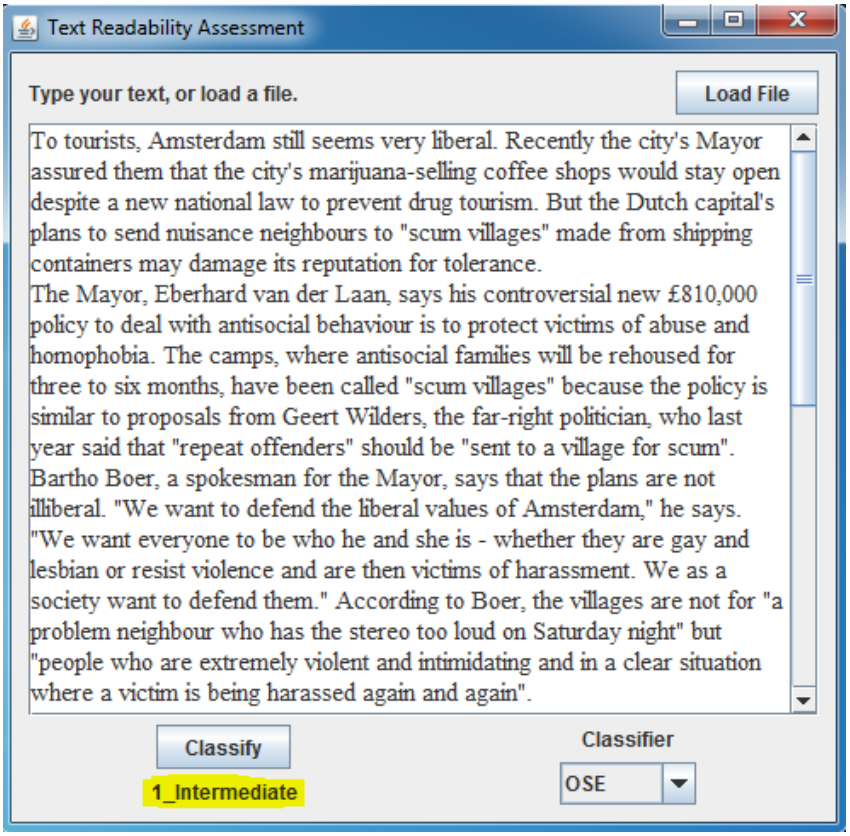
2.5 واجهة التطبيق ودليل استخدامها

يبيّن الشكل 2.5 الواجهة التخاطبية للتطبيق. يمكن للمستخدم إدخال نص بشكل مباشر باستخدام لوحة المفاتيح أو تحميل ملف نصي عبر الزر Load File. وبالضغط على الزر Classify يظهر التطبيق النتيجة. قد تستغرق هذه العملية زمن، حوالي 12–4 ثانية، بحسب طول وتعقيد النص المدخل. ويمكن للمستخدم اختيار مُصنّف من القائمة الموجودة تحت اسم Classifier (الزاوية اليمنى من الأسفل). لمعرفة سبب سماح التطبيق للمستخدم بتحديد المصنّف الذي سيتم استخدامه، انظر إلى بداية الفقرة 1.3.



الشكل 2.5: الواجهة التخاطبية للتطبيق.

يبيّن الشكل 3.5 مثال على استخدام التطبيق لتقييم مقروئية النص Amsterdam-int.txt، الموجود ضمن مجموعة المعطيات OSE. ونلاحظ أن التطبيق قام بتصنيفه بشكل صحيح كنص متوسط الصعوبة.



الشكل 3.5: مثال على استخدام التطبيق.

الفصل السادس

التقييم والنتائج

يبيّن هذا الفصل النتائج التي حصلنا عليها بعد التنجيز والاختبار. ويوضح أهم الملاحظات حول هذه النتائج.

1.6 نتائج الـ OSE

إنّ أفضل نسبة صحّة تم تحقيقها على مجموعة المعطيات هذه هي 78.13%، باستخدام 155 سمة [26]. وللأسف لم يتم ذكر معايير أخرى لتقييم نتائجهم غير نسبة الصحّة. نسبة الصحّة التي تم تحقيقها في هذا المشروع هي 80.83%. وذلك باستخدام 30 سمة فقط. جميع النتائج المذكورة تخص مجموعة الاختبار المستخدمة. حيث كما ذكرنا سابقاً، يوجد 567 مثال تدريب. تم فصل هذه العينات بنسبة 66%. تم استخدام القسم الأكبر منها كمجموعة التدريب، واستخدام القسم الأصغر كمجموعة الاختبار.

يبيّن الجدول 1.6 قيم المعايير المختلفة المستخدمة لتقييم الأداء. لمعرفة دلالاتها انظر إلى الفقرة 4.1.2. إن مجمل النتائج جيد. حيث نلاحظ أن جميع هذه القيم تتعدى الـ 0.7. وإن القيم المتوسطة تتعدى الـ 0.8. ونلاحظ أن المستوى الذي نقوم بتمييزه بالشكل الأفضل هو المستوى المبتدئ. والمستوى الذي نقوم بتمييزه بالشكل الأسوء هو المستوى المتوسط، وهذا متوقع كون المستوى المتوسط يقع بين المستويين المبتدئ والمتقدم مما يجعل احتمال الخطأ بتصنيفه أكبر.

يبيّن الجدول 2.6 مصفوفة الخطأ للنتائج. نلاحظ أن التمييز بين المستويين المبتدئ والمتقدم يتم بدون خطأ.

ونلاحظ أنه تم تصنيف عينات من المستوى المتوسط على أنها من المستوى المتقدم ومن المستوى المبتدئ بالتساوي (الرقم 10 في الجدول). الخطأ الأكبر في التصنيف هو تصنيف عينات من المستوى المتقدم على أنها من المستوى المتوسط (الرقم 13 في الجدول). قد يعود ذلك لكون الفارق بين المستويين المتوسط والمبتدئ أكبر من الفارق بين المستويين المتوسط والمتقدم.

Level	Precision	Recall	F-Score
Elementary	0.865	0.941	0.901
Intermediate	0.742	0.710	0.726
Advanced	0.811	0.768	0.789
Average	0.806	0.808	0.806

جدول 1.6: معايير تقييم الأداء لمعطيات الـ OSE.

actual \ predicted	Elementary	Intermediate	Advanced
Elementary	64	4	0
Intermediate	10	49	10
Advanced	0	13	43

جدول 2.6: مصفوفة الخطأ لمعطيات الـ OSE.

الفصل السابع

خاتمة المشروع

يختتم هذا الفصل المشروع. فيبيّن الفائدة المكتسبة من المشروع، والمشكلات والصعوبات التي واجهتنا خلاله. ويبيّن بعض الآفاق المستقبلية.

1.7 الخاتمة والفائدة المكتسبة

في هذا المشروع، تم بناء تطبيق لتقييم مقروئية نصوص اللغة الانكليزية. بالإضافة إلى تنجيز مكتبة سهلة الاستخدام وقابلة للتوسع لاستخراج السمات من النصوص. تم التعرف على عدد واسع من المفاهيم والطرائق خلال تنفيذ هذا المشروع. فتم التعرف على مفاهيم تعلم الآلة المختلفة لأول مرّة، بدءاً من مرحلة تنظيف المعطيات، وحتى مرحلة استخدام النموذج الناتج في تطبيق نهائي. والتعرف على عدد من الأدوات المفيدة لتطبيقها بشكل عملي. كما تم التعرف على مفاهيم معالجة اللغات الطبيعية ومختلف الأدوات المتوفرة لها. وتم بناء كود برمجي ليس بحجم صغير مما أدى إلى اكتساب خبرات في تصميم المكونات البرمجية والتعرف على أنماط تصميمية جديدة.

2.7 المشكلات والصعوبات

الصعوبات الأساسية التي واجهتنا في المشروع تتمثل بـ:

- الحاجة إلى دراسة مرجعية مطوّلة كوننا نتعامل مع مفاهيم تعلم الآلة لأول مرة.
- تنوّع المهام ضمن المشروع. فتوجد عدّة مراحل ضمن المشروع، كل منها يحتاج وقت وجهد. مثل الحاجة إلى تجهيز المعطيات وتنظيفها. والحاجة إلى كتابة كود برمجي كبير نسبياً لاستخراج مجموعة واسعة من السمات التي تختلف طرائق حسابها. والحاجة إلى دراسة عدة أدوات ومعرفة آلية استخدامها.
- صعوبة الموضوع المدروس. إن عملية تقييم مقروئية نص ليست بأمر سهل. ومعايير التقييم ليست دقيقة. فقد يتدخل فيها طابع شخصي؛ أي يختلف التقييم باختلاف الشخص. وقد يسبب ذلك عدم دقّة في المعطيات التي تم استخدامها للتدريب.

3.7 آفاق مستقبلية

توجد العديد من الآفاق المستقبلية التي يمكن طرحها بناءً على هذا المشروع. فيمكن تطوير هذا المشروع بتمديده إلى لغات أخرى. ويمكن توسيع التطبيق النهائي بإضافة مُصنفات أخرى لتوسيع قابلية استخدامه. ويمكن إضافة مجموعة جديدة من السمات وتحسين أداء عملية التصنيف. ويمكن تفصيل التقييم الكلي إلى عدّة جوانب، مثل إظهار تقييم للمفردات، وإظهار تقييم للصياغة، وهكذا.

أيضاً بالإمكان استخدام هذا المشروع كمكوّن جزئي ضمن نظام أوسع. فيمكن إضافته ضمن محرر نصوص أو ضمن المتصفح، بحيث يمكن للمستخدم تحديد نص وتقييم مقروئيته. مما سيزيد من قدرات محررات النصوص الحالية، ويجعل عملية الكتابة أسهل. ومن الممكن إضافته ضمن محركات البحث بطريقة تسمح للمستخدم بالبحث عن مقال معين وبصعوبة معينة.

المراجع

[1] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. "Building a large annotated corpus of English: The Penn Treebank". In: *Computational linguistics* 19.2 (1993), pp. 313–330.

[2] Averil Coxhead. "A new academic word list". In: *TESOL quarterly* 34.2 (2000), pp. 213–238.

[3] Sarah E Schwarm and Mari Ostendorf. "Reading level assessment using support vector machines and statistical language models". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 2005, pp. 523–530.

[4] Rong Zheng et al. "A framework for authorship identification of online messages: Writing-style features and classification techniques". In: *Journal of the American society for information science and technology* 57.3 (2006), pp. 378–393.

[5] William H DuBay. "The Classic Readability Studies." In: *Online Submission* (2007).

[6] Ronald P Reck and Ruth A Reck. "Generating and rendering readability scores for Project Gutenberg texts". In: *Proceedings of the Corpus Linguistics Conference*. 2007.

[7] Marie-Catherine De Marneffe and Christopher D Manning. *Stanford typed dependencies manual*. Tech. rep. Technical report, Stanford University, 2008.

[8] Lijun Feng, Noémie Elhadad, and Matt Huenerfauth. "Cognitively motivated features for readability assessment". In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2009, pp. 229–237.

[9] Sarah E Petersen and Mari Ostendorf. "A machine learning approach to reading level assessment". In: *Computer speech & language* 23.1 (2009), pp. 89–106.

[10] Lijun Feng. "Automatic readability assessment". In: (2010).

[11] Lijun Feng et al. "A comparison of features for automatic readability assessment". In: *Proceedings of the 23rd international conference on computational linguistics: Posters*. Association for Computational Linguistics. 2010, pp. 276–284.

[12] Xiaofei Lu. "Automatic analysis of syntactic complexity in second language writing". In: *International journal of corpus linguistics* 15.4 (2010), pp. 474–496.

[13] Scott A Crossley, David B Allen, and Danielle S McNamara. "Text readability and intuitive simplification: A comparison of readability formulas." In: *Reading in a foreign language* 23.1 (2011), pp. 84–101.

[14] Scott A Crossley, David Allen, and Danielle S McNamara. "Text simplification and comprehensible input: A case for an intuitive approach". In: *Language Teaching Research* 16.1 (2012), pp. 89–108.

[15] Sowmya Vajjala and Detmar Meurers. "On improving the accuracy of readability classification using insights from second language acquisition". In: *Proceedings of the seventh workshop on building educational applications using NLP*. Association for Computational Linguistics. 2012, pp. 163–173.

[16] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[17] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[18] Sowmya Vajjala and Detmar Meurers. "Readability assessment for text simplification: From analysing documents to identifying sentential simplifications". In: *ITL-International Journal of Applied Linguistics* 165.2 (2014), pp. 194–222.

[19] D Sculley et al. "Hidden technical debt in machine learning systems". In: *Advances in neural information processing systems*. 2015, pp. 2503–2511.

[20] Sowmya Vajjala. "Analyzing text complexity and text simplification: connecting linguistics, processing and educational applications". PhD thesis. Ph. D. thesis, University of Tübingen, 2015.

[21] Sowmya Vajjala and Detmar Meurers. "Readability-based sentence ranking for evaluating text simplification". In: *arXiv preprint arXiv:1603.06009* (2016).

[22] Ian H Witten et al. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[23] Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. "Text readability assessment for second language learners". In: *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*. 2016, pp. 12–22.

[24] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2017.

[25] Riad Sonbol. *Extracting Business Process Models from Natural Language Texts*. Higher Institute for Applied Sciences and Technology, 2017.

[26] Sowmya Vajjala and Ivana Lucic. "OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification". In: (2018).