2013

Information Retrieval Project

بإشراف الدكتور سامي الخيمي

تقديم الطلاب

أسامة سليق

آنا عجميان

بلال الهلال الشريفي

لما موازيني

1 مقدمة

نظم استرجاع المعلومات هو الفرع من علوم الحاسب الذي يركز على كسب، تنظيم، تخزين، استرجاع، وتوزيع المعلومات.

يرتكز الهدف الأساسي من نظم استرجاع المعلومات على مساعدة المستخدمين على إيجاد المعلومات التي تلبي احتياجاتهم وتساؤلاتهم.

أصبحت نظم استرجاع المعلومات محط تركيز نظريات الإنترنت والتقنيات والتطبيقات التي تعتمد عى معالجة كمية كبيرة من البيانات.

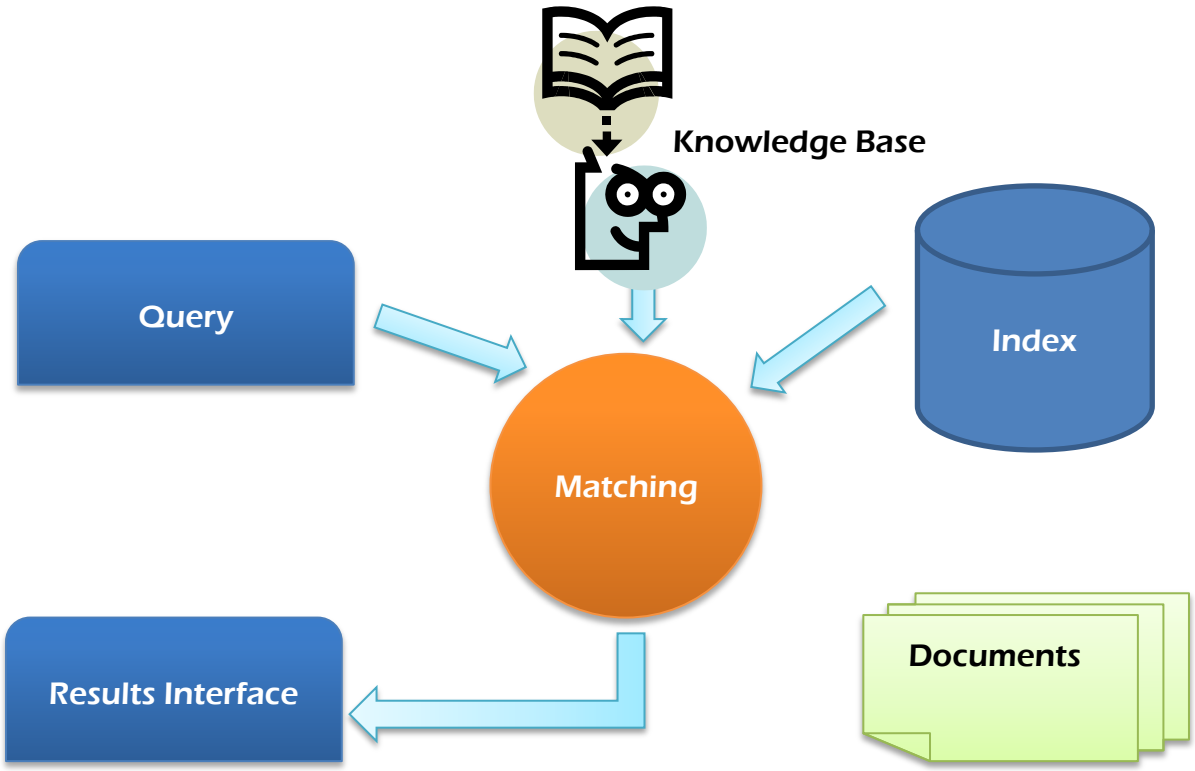
2 هيكلية نظام استرجاع المعلومات

- نظام استرجاع المعلومات يهتم بقضايا تمثيل المعلومات "كيف يمكن تمثيل المعلومات التي نريد؟؟"، تخزين المعلومات "كيف سيتم تخزين المعلومات وأين؟؟"، تنظيم المعلومات "كيف سيتم تنظيم المعلومات؟؟"، والوصول إلى هذه المعلومات بعد عمليات المقارنة "Matching Evaluation"؛
- **والسؤال المهم هنا هو كيف سيتم تمثيل عنصر المعلومات Information Item؟؟.**

إن خصوصية نظام استرجاع المعلومات هنا مرتبطة بشكل أساسي بطبيعة ال Information Item، فعندما يكون نظام استرجاع المعلومات يعتمد على قاعدة معطيات فإن عنصر المعلومات في هذه الحالة هو سجل، وعندما يكون نظام استرجاع المعلومات معتمد على صفحات الويب فإن عنصر المعلومات هنا سيكون وثيقة؛

- في نظام استرجاع المعلومات لا بد من تمثيل المحتوى الممثل بعناصر المعلومات وتسمى هذه العملية بفهرسة الوثائق، وتمثيل محتوى الاستعلام "ما هو الاستعلام المستخدم في البحث، كيف سيتم تمثيله، وماهو النموذج الذي سيبنى عليه"، ومن ثم تتم عمليات المقارنة بين نموذج عناصر المعلومات Document Model ونموذج الاستعلام Query Model وينتج عن عمليات المقارنة هذه نتائج الاستعلام ضمن فضاء المعلومات المستخدم.

- يوضح الشكل آلية عمل نظام استرجاع المعلومات:



3 متطلبات النظام

المرحلة الأولى		
رقم المهمة	المهمة	تاريخ الانتهاء
1	بناء الوكيل البرمجي المسؤول عن جمع بيانات الأفلام.	5/4/2013
2	بناء التطبيق المسؤول عن قراءة نص وتحويله إلى RDF.	15/4/2013
3	التحقق من صفوف مجموعات البيانات في المخزن الشخصي الذي تم إنشاؤه.	19/4/2013
المرحلة الثانية		
1	تحسين صفوف البيانات وبناء نموذج أولي للواجهات الممكن استخدامها لهذا الغرض.	27/4/2013
2	تحقيق آلية للاستعلام عن المعلومات في المخزن الشخصي الذي تم إنشاؤه لصفوف البيانات.	30/4/2013
3	ربط المخزن الشخصي الذي تم إنشاؤه بأحد المخازن العامة مثل DBPedia.	10/5/2013
4	بناء واجهات برمجية لعملية البحث من خلال الكلمات المفتاحية أو من خلال exploratory Navigation	19/5/2013

4 بناء الوكيل البرمجي المسؤول عن جمع بيانات الأفلام

- يقوم هذا الوكيل البرمجي بعمليات التصفح عبر مجموعة من المواقع يحدَدها مستخدم النظام، والقيام بجمع ملفات subtitles المتوفرة و المرتبطة بوصلات الأفلام المتوفرة في الصفحة، وتخزينها وفق مسارات وجودها ضمن مجلدات الأفلام؛
- إذاً دخل هذا الوكيل موقع أو مجموعة من مواقع الأفلام؛
- إذ يقوم مستخدم النظام بتحديد مجموعة من المواقع التي يريد تحميل نصوص الترجمة منها؛
- يتسلم كل موقع من المواقع المختارة thread ليقوم بعمليات المعالجة والتحميل؛
- من أجل كل رابط يتم الحصول على كل الروابط الموجودة ضمنها وتخزينها ضمن قائمة لتتم عمليات المعالجة عليها تدريجيا؛
- في حال كان أحد الروابط المعالجة هو رابط لملف ترجمة يتم التحقق من حجم الملف المراد تحميله، ففي حال كان أكبر من 200KB لن تتم عملية تحميل هذا الرابط، وإلا يتم تحميل نص الترجمة.
- من أجل كل صفحة تحوي رابط تحميل نص ترجمة يتم أخذ النصوص الموجودة ضمن الوسومات title, heading1 للحصول على اسم الفيلم؛
- من أجل كل نص موجود ضمن الوسومات السابقة يتم البحث عنها ضمن قاعدة معطيات الأفلام IMDB وقاعدة المعطيات TMDB والتي تقدمان خدمة البحث ضمنها من خلال web service متوفرة على شبكة الإنترنت، حيث يتم مقارنة النص مع أسماء الأفلام الموجودة ضمن قاعدتي المعطيات هذه والحصول على اسم الفيلم في حال كانت هذه الوسومات تحمل اسم فيلم؛
- معرفة لغة نص الترجمة الذي تم تحميله من خلال إحصائيات عن اللغات ومعالجة النص المدخل ومقارنته مع هذه الإحصائيات؛

- خرج الوكيل مجموعة من ملفات نصوص الترجمة الموجودة ضمن الموقع ضمن مسارات محددة وتخزين معلوماتها ضمن قاعدة معطيات خاصة بالأفلام.

4.1 التحسينات التي قمنا بتطبيقها على الوكيل البرمجي

4.1.1 Language Detection

- اكتشاف لغة الملف الذي نقوم بتحميله اعتمادا على إحصائيات تضم أشهر Terms ضمن اللغات، حيث يتم بناء نموذج خاص بالنص المدخل يحوي terms النص المدخل مع عدد تكرار كل term ضمن النص، كما يتم بناء نموذج خاص باللغة المقارن بها، ويتم حساب المسافة بين هذين النموذجين.
- يتم تكرار هذه العملية من أجل مجموعة من اللغات المحتملة، وبالتالي يتم احتساب المسافة بين نموذج النص المدخل ونموذج كل لغة من اللغات المحتملة، وفي النهاية يتم اعتبار أصغر مسافة، وبالتالي يتم الحصول على اللغة التي كتب فيها النص المدخل.
- languageStatistics تضم هذه الإحصائيات كل لغة ومجموعة من terms الأساسية الموجودة فيها، تشمل هذه الإحصائيات حوالي 65 لغة.

4.1.2 Multithreading

يتسلم كل thread موقع من المواقع التي اختارها مستخدم النظام ليقوم كل thread على حدا بعمليات المعالجة والتحميل.

4.1.3 IMDB

الحصول على اسم الفيلم من خلال فحص مجموعة من النصوص الموجودة ضمن وسومات خاصة ضمن الصفحة من المحتمل وجود اسم الفيلم ضمنها، حيث يقوم الوكيل البرمجي بالاتصال بقاعدة معطيات ضخمة تدعى IMDB والتي تقدم web service متاحة على شبكة الإنترنت، وفحص هذه النصوص فيما إذا كانت تحوي اسم فيلم أم لا، وفي حال الإيجاب نكون قد حصلنا على اسم الفيلم الموجود ضمن صفحة رابط التحميل.

TMDB 4.1.4

الحصول على اسم الفيلم من خلال فحص مجموعة من النصوص الموجودة ضمن وسومات خاصة ضمن الصفحة من المحتمل وجود اسم الفيلم ضمنها، حيث يقوم الوكيل البرمجي بالاتصال بقاعدة معطيات ضخمة تدعى TMDB والتي تقدم web service متاحة على شبكة الإنترنت، وفحص هذه النصوص فيما إذا كانت تحوي اسم فيلم أم لا، وفي حال الإيجاب نكون قد حصلنا على اسم الفيلم الموجود ضمن صفحة رابط التحميل.

ملاحظة

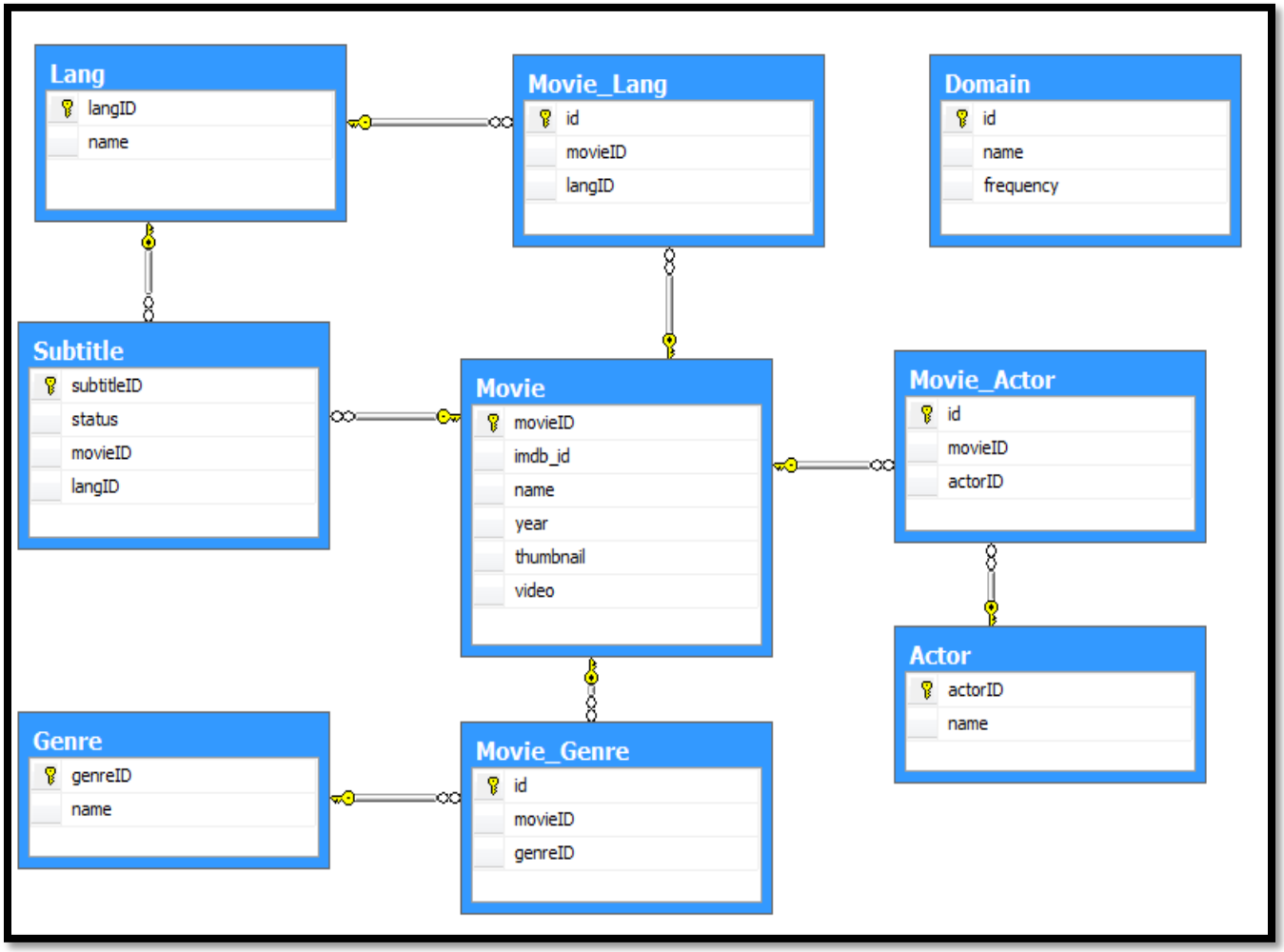
تم الاستعانة بقاعدتي معطيات لمصادفتنا حالات وجود اسم الفيلم ضمن قاعدة المعطيات IMDB وعدم وجودها ضمن قاعدة المعطيات TMDB، وفي بعض الأحيان نجد اسم الفيلم ضمن TMDB ولا يكون موجودا ضمن IMDB لذلك تم جمع نتائج البحث ضمن الخدمتين والحصول على اسم الفيلم المطلوب.

File Size 4.1.5

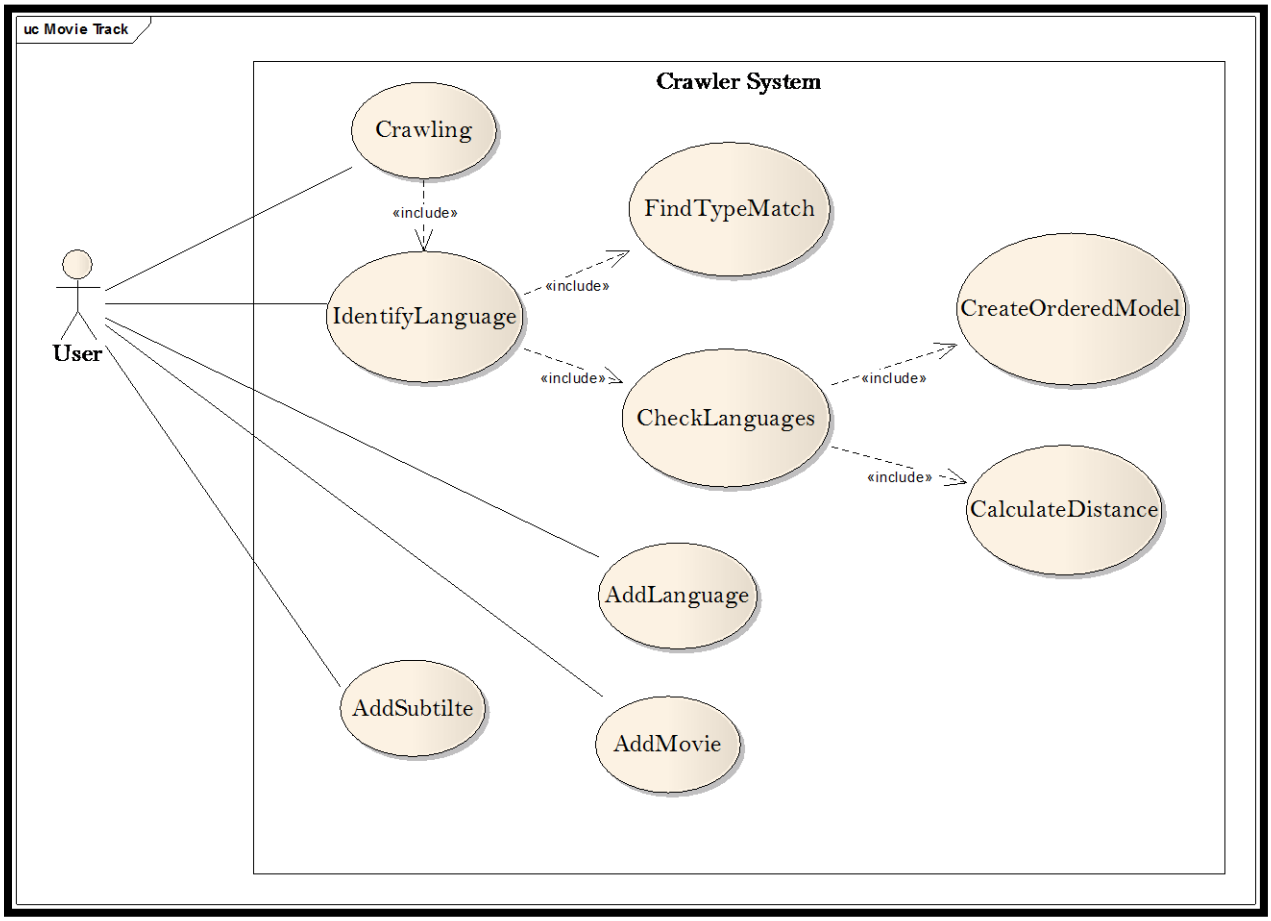
فحص حجم الملف قبل تحميله ففي حال كان أكبر من حجم معين لا يقوم الوكيل البرمجي بتحميله كونه من غير الممكن أن يكون ملف ترجمة.

- يتم تخزين الأفلام التي يتم إيجادها أثناء تصفح المواقع في قاعدة معطيات Movie Track؛
- من أجل كل فيلم يتم تخزين ملفات الترجمة ضمن قاعدة المعطيات؛

4.2 مخطط قاعدة المعطيات المستخدمة لتخزين نتائج الوكيل البرمجي



4.3 حالات الاستخدام الخاصة بمرحلة بناء الوكيل البرمجي



حالة الاستخدام	FindTypeMatch
رقم حالة الاستخدام	1
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	إيجاد عدد التطابقات بين أنماط الترميز والنص المدخل.
الشروط المسبقة	لايوجد.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام معرفة عدد التطابقات بين كل نمط من أنماط الترميز والنص المدخل؛ 2- من أجل كل نمط من الأنماط الموجودة يتم احتساب عدد التقاطعات بينه وبين النص المدخل؛ 3- يتم احتساب نسبة التقارب من خلال قسمة عدد التقاطعات على طول النص المدخل.
الشروط اللاحقة	تم الحصول على قائمة بعدد التقاطعات مع النص المدخل من أجل كل نمط ترميز.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

حالة الاستخدام	CreateOrderedModel
رقم حالة الاستخدام	2
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	بناء النموذج الخاص بالنص المدخل والذي سيتم الاعتماد عليه في معرفة لغة النص المدخل.
الشروط المسبقة	لايوجد.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام بناء النموذج الخاص بالنص الذي يريد معرفة اللغة الخاصة به؛ 2- تقسيم النص المدخل إلى ثلاثيات؛ 3- في حال كانت الثلاثية الحالية موجودة من قبل يتم زيادة عدد مرات تكرارها بمقدار واحد؛ 4- في حال لم تكن الثلاثية الحالية موجودة من قبل يتم جعل عدد مرات تكرارها يساوي الواحد؛
الشروط اللاحقة	الحصول على قائمة تضم terms النص المدخل مع عدد مرات تكرار كل منها.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

حالة الاستخدام	CalculateDistance
رقم حالة الاستخدام	3
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	حساب المسافة بين نموذج النص المدخل ونموذج لغة معروفة.
الشروط المسبقة	لايوجد.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام حساب المسافة بين نموذج النص ونموذج لغة ما؛ 2- من أجل كل term من terms النص المدخل في حال كان ينتمي إلى terms اللغة المعروفة يتم زيادة عدد يمقدار عدد مرات تكرار هذا ال term ضمن النص مطروحا منه قيمة هذا ال term ضمن النموذج المعروف إلى قيمة المسافة بين النموذجين؛ 3- في حال لم يكن ينتمي إلى اللغة المعروفة يتم زيادة عدد بمقدار 300 إلى قيمة المسافة بين النموذجين.
الشروط اللاحقة	الحصول على المسافة بين النموذج النص المدخل ونموذج لغة معروفة.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

حالة الاستخدام	CheckLanguages
رقم حالة الاستخدام	4
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	إيجاد اللغة التي قد كتب فيها النص المدخل.
الشروط المسبقة	لايوجد.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام معرفة اللغة التي يتبع لها النص المدخل؛ 2- بناء النموذج الخاص بالنص المدخل لمقارنته مع نماذج اللغات Include CreateOrderedModel 3- من أجل كل لغة يتم بناء النموذج الخاص بها وحساب المسافة بينه وبين نموذج النص المدخل Include CalculateDistance 4- اختيار اللغة المقابلة لأصغر مسافة بين نموذجها ونموذج النص المدخل.
الشروط اللاحقة	الحصول على لغة النص المدخل.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

حالة الاستخدام	IdentifyLanguage
رقم حالة الاستخدام	5
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	اكتشاف لغة النص المدخل.
الشروط المسبقة	لايوجد.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام اكتشاف لغة النص المدخل؛ 2- معرفة عدد التطابقات بين النص المدخل وأنماط الترميز Include FindTypeMatch 3- محاولة معرفة اللغة من خلال قواعد متعارف عليها إحصائيا؛ 4- في حال لم يتم التعرف على اللغة من خلال هذه القواعد يتم حساب المسافة بين نموذج لغات محددة ونموذج النص المدخل Include CheckLanguages
الشروط اللاحقة	تم الحصول على لغة النص المدخل.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

حالة الاستخدام	AddLanguage
رقم حالة الاستخدام	6
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	إضافة لغة جديدة إلى قائمة اللغات الخاص بنصوص ترجمة الأفلام.
الشروط المسبقة	لايوجد.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام إضافة لغة جديدة إلى قائمة اللغات؛ 2- يتم التحقق من كون اللغة الجديدة غير موجودة مسبقا ضمن قائمة اللغات وفي حال الإيجاب يتم إضافة اللغة الجديدة إلى قائمة اللغات.
الشروط اللاحقة	تم إضافة لغة جديدة إلى القائمة.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

حالة الاستخدام	AddMovie
رقم حالة الاستخدام	7
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	إضافة فيلم جديد إلى قائمة الأفلام.
الشروط المسبقة	لايوجد.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام إضافة فيلم جديد إلى قائمة الأفلام؛ 2- يتم التحقق من كون الفيلم الجديد غير موجود ضمن قائمة الأفلام، في حال الإيجاب يتم إضافة الفيلم الجديد إلى القائمة.
الشروط اللاحقة	تم إضافة فيلم جديد إلى القائمة.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

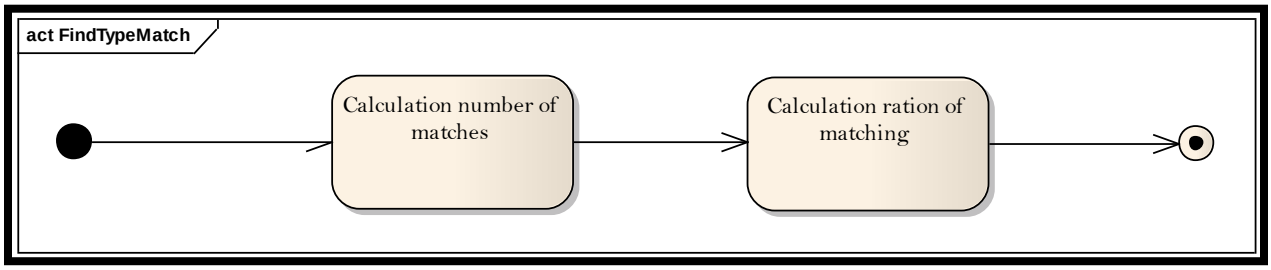
حالة الاستخدام	AddSubtilte
رقم حالة الاستخدام	8
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	إضافة نص ترجمة جديد إلى قائمة نصوص الترجمة الخاصة بفيلم ما.
الشروط المسبقة	لايوجد.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام إضافة نص ترجمة إلى قائمة نصوص الترجمة التابعة لفيلم ما؛ 2- يتم التحقق من كون لغة النص الجديد موجودة ضمن قائمة اللغات، في حال لم تكن موجودة يتم إضافة اللغة أولا إلى قائمة اللغات المتاحة؛ 3- يتم التحقق من كون فيلم نص الترجمة موجود ضمن قائمة الأفلام، في حال لم يكن موجود يتم إضافة الفيلم أولا إلى قائمة الأقلام؛ 4- يتم التحقق من أن نص الترجمة موجود مسبقا، في حال لم يكن موجود يتم إضافته إلى قائمة نصوص الترجمة الخاصة بفيلم ما؛ 5- إضافة نص الترجمة إلى قائمة نصوص الترجمة.
الشروط اللاحقة	تم إضافة نص ترجمة جديد إلى قائمة النصوص الخاصة بفيلم ما.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

حالة الاستخدام	Crawling
رقم حالة الاستخدام	9
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	البحث عن نصوص الترجمة ضمن مجموعة من المواقع وتحميلها.
الشروط المسبقة	لايوجد.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام البحث عن نصوص الترجمة ضمن مجموعة من المواقع المحددة مسبقا وتحميلها؛ 2- يقوم مستخدم النظام بتحديد مجموعة من المواقع التي يريد تحميل نصوص الترجمة منه؛ 3- من أجل كل رابط يتم الحصول على كل الروابط الموجودة ضمنها وتخزينها ضمن قائمة لتتم عمليات المعالجة عليها تدريجيا؛ 4- في حال كان أحد الروابط المعالجة هو رابط لملف ترجمة يتم التحقق من حجم الملف المراد تحميله، ففي حال كان أكبر من 200KB لن تتم عملية تحميل هذا الرابط، وإلا يتم تحميل نص الترجمة. 5- من أجل كل صفحة تحوي رابط تحميل نص ترجمة يتم أخذ النصوص الموجودة ضمن الوسومات title, heading1 للحصول على اسم الفيلم؛ 6- من أجل كل نص موجود ضمن الوسومات السابقة يتم البحث عنها ضمن قاعدتي معطيات الأفلام IMDB و TMDB، حيث يتم مقارنة النص مع

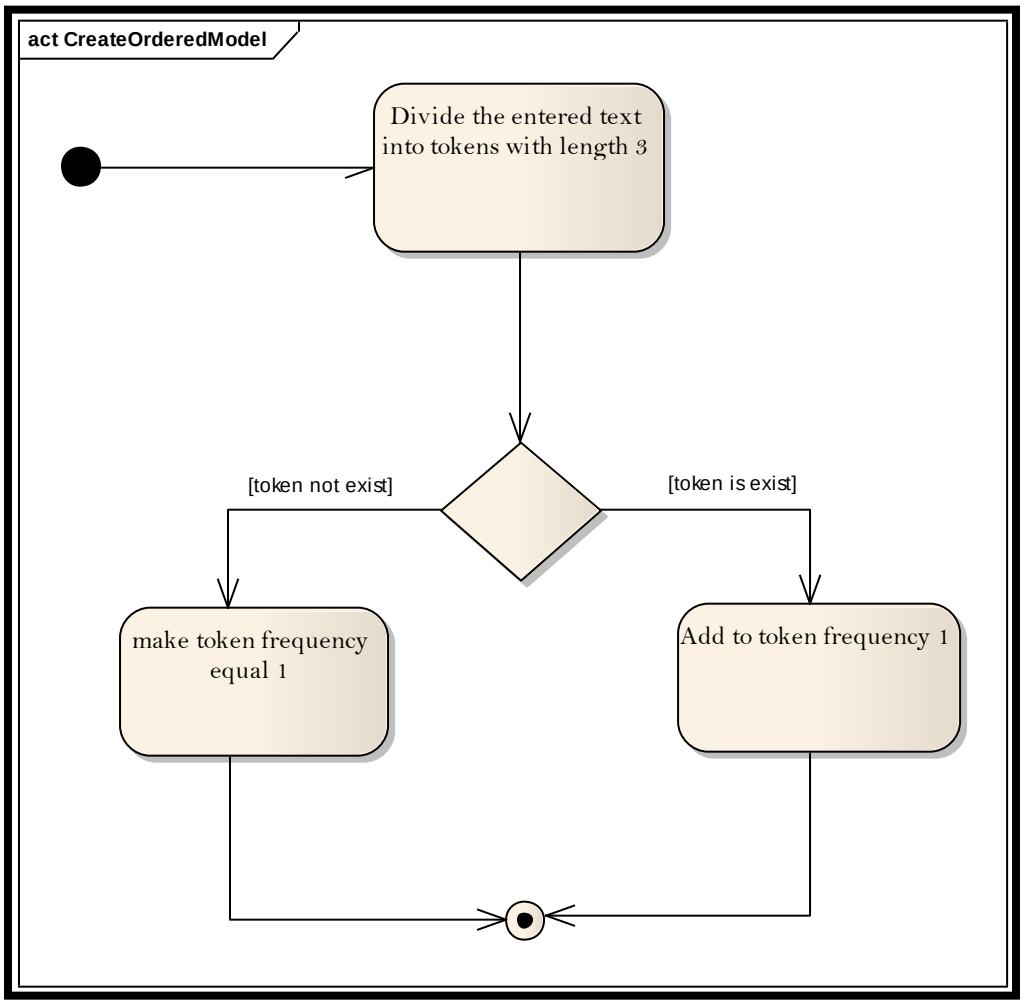
	أسماء الأفلام الموجودة ضمن قاعدتي المعطيات هذه والحصول على اسم الفيلم في حال كانت هذه الوسومات تحمل اسم فيلم؛ 7- معرفة لغة نص الترجمة الذي تم تحميله Include IdentifyLanguage 8- التخزين ضمن قاعدة المعطيات الخاصة بالأفلام لكل من اللغة والفيلم ونص الترجمة.
الشروط اللاحقة	تم البحث عن ملفات نصوص الترجمة وتحميلها ضمن مسارات محددة.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

4.4 مخططات الأنشطة Activity Diagrams الخاصة بمرحلة بناء الوكيل البرمجي

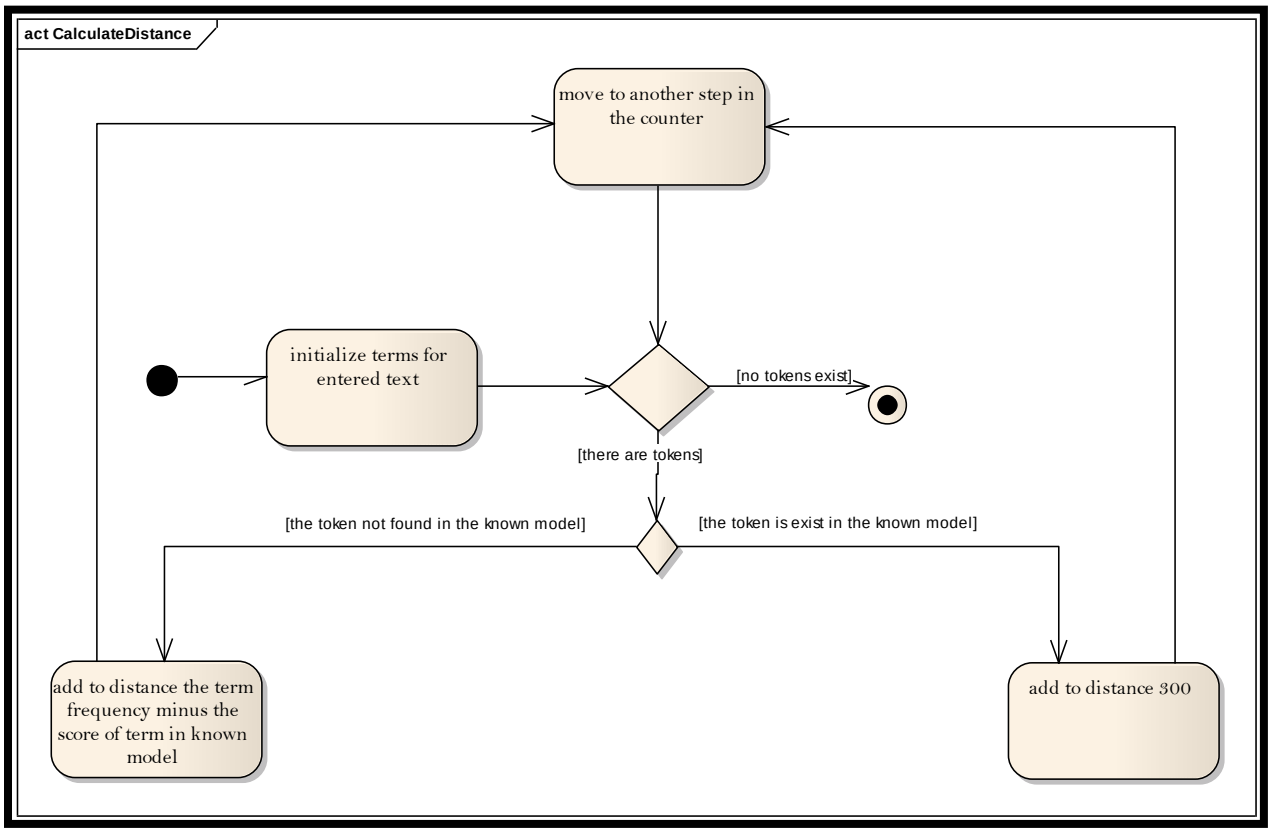
FindTypeMatch –



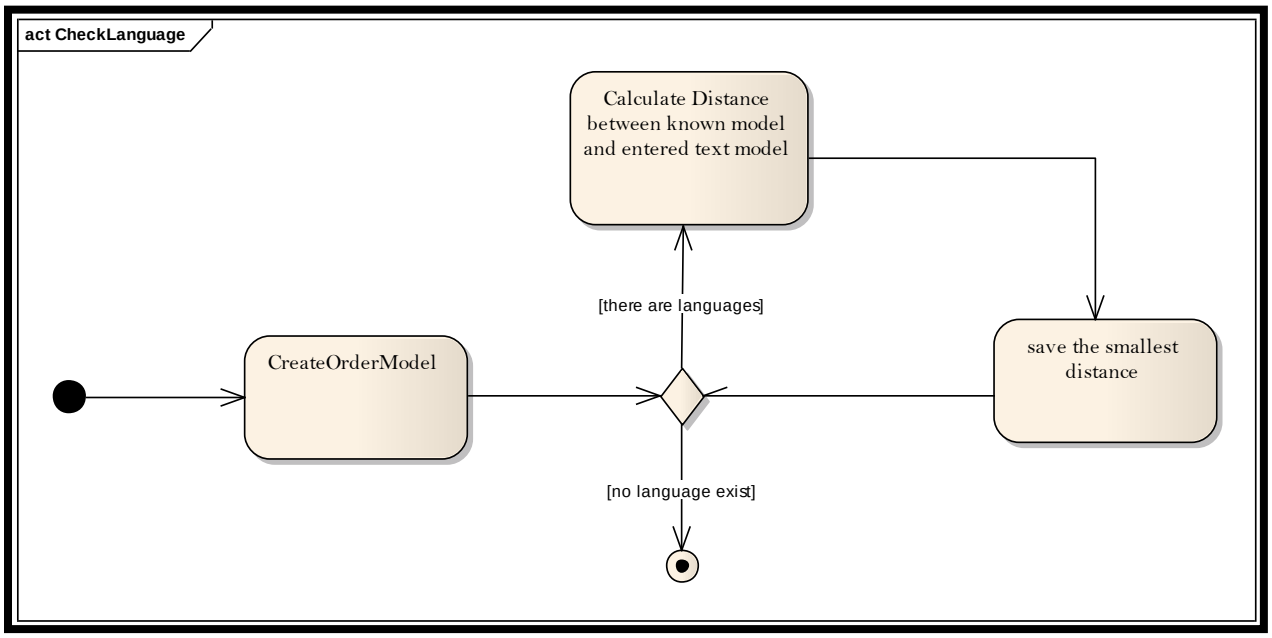
CreateOrderedModel –



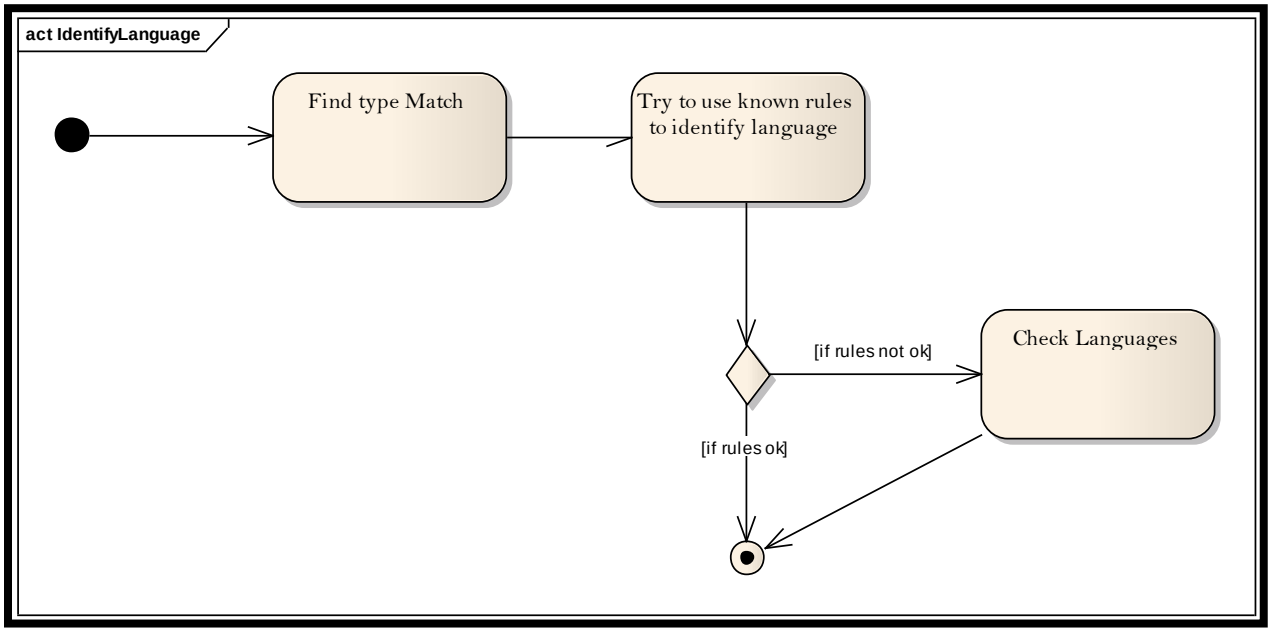
CalculateDistance –



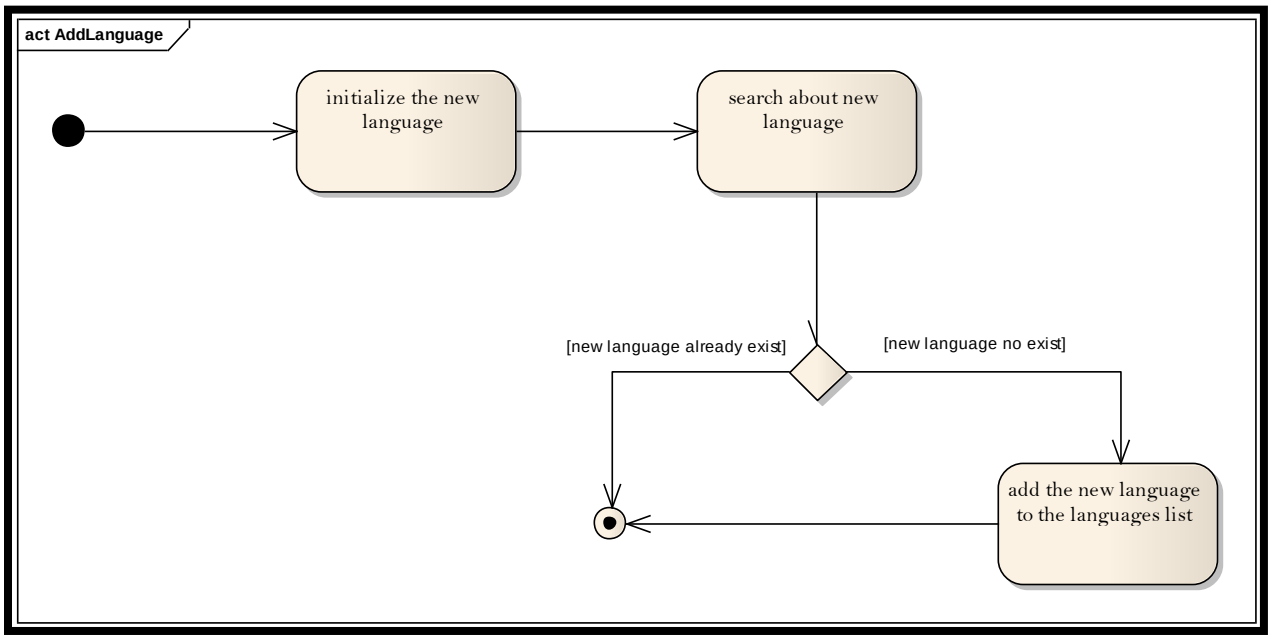
CheckLanguages –



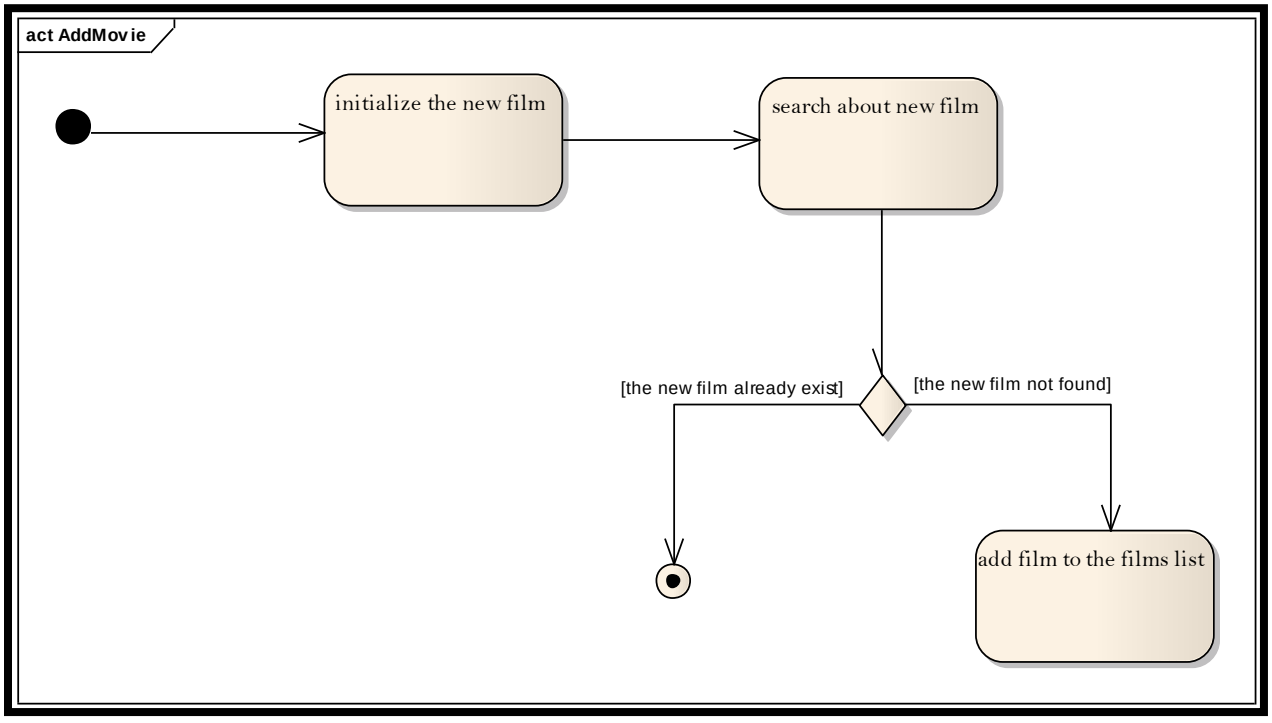
IdentifyLanguage –



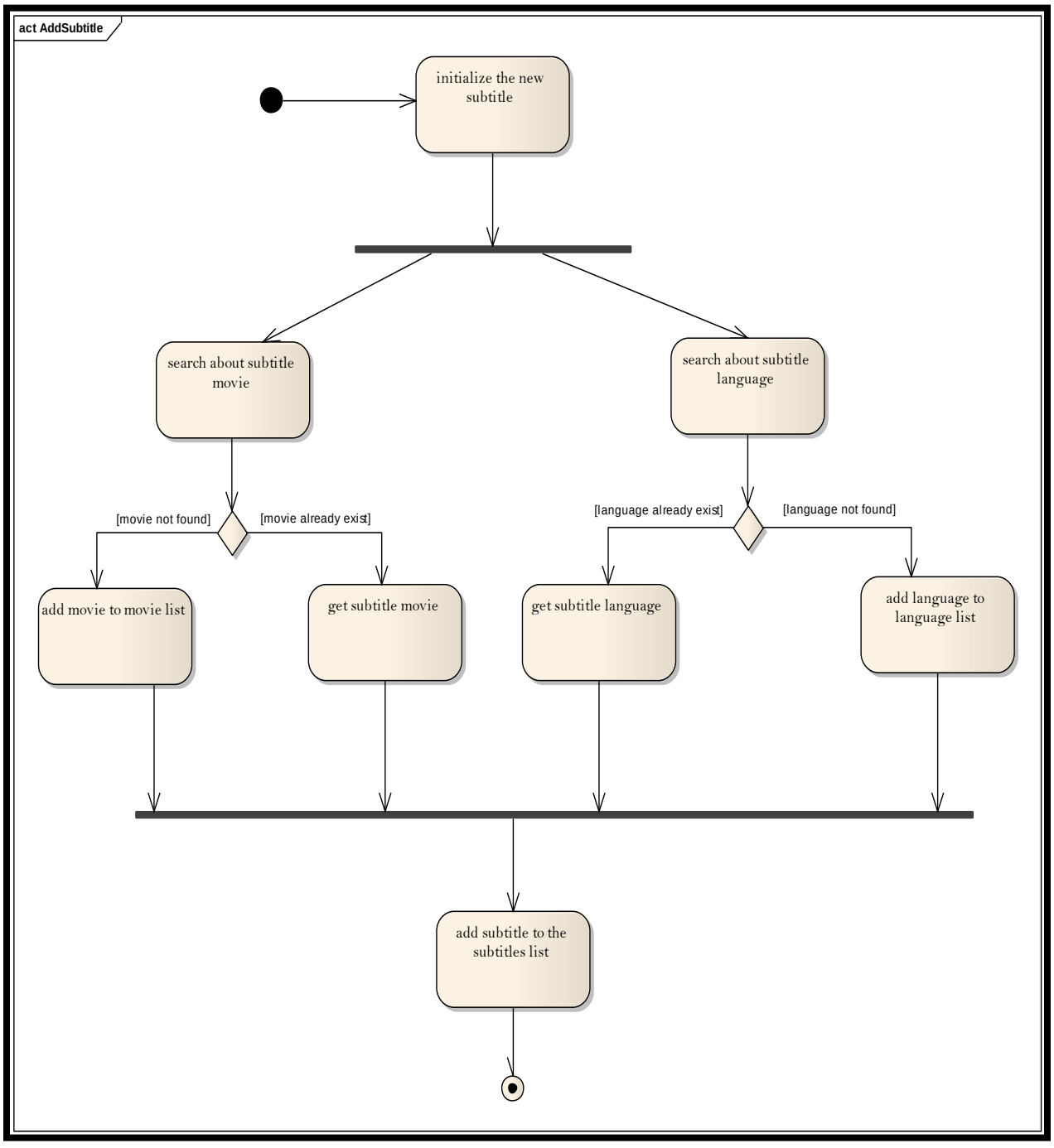
AddLanguage –



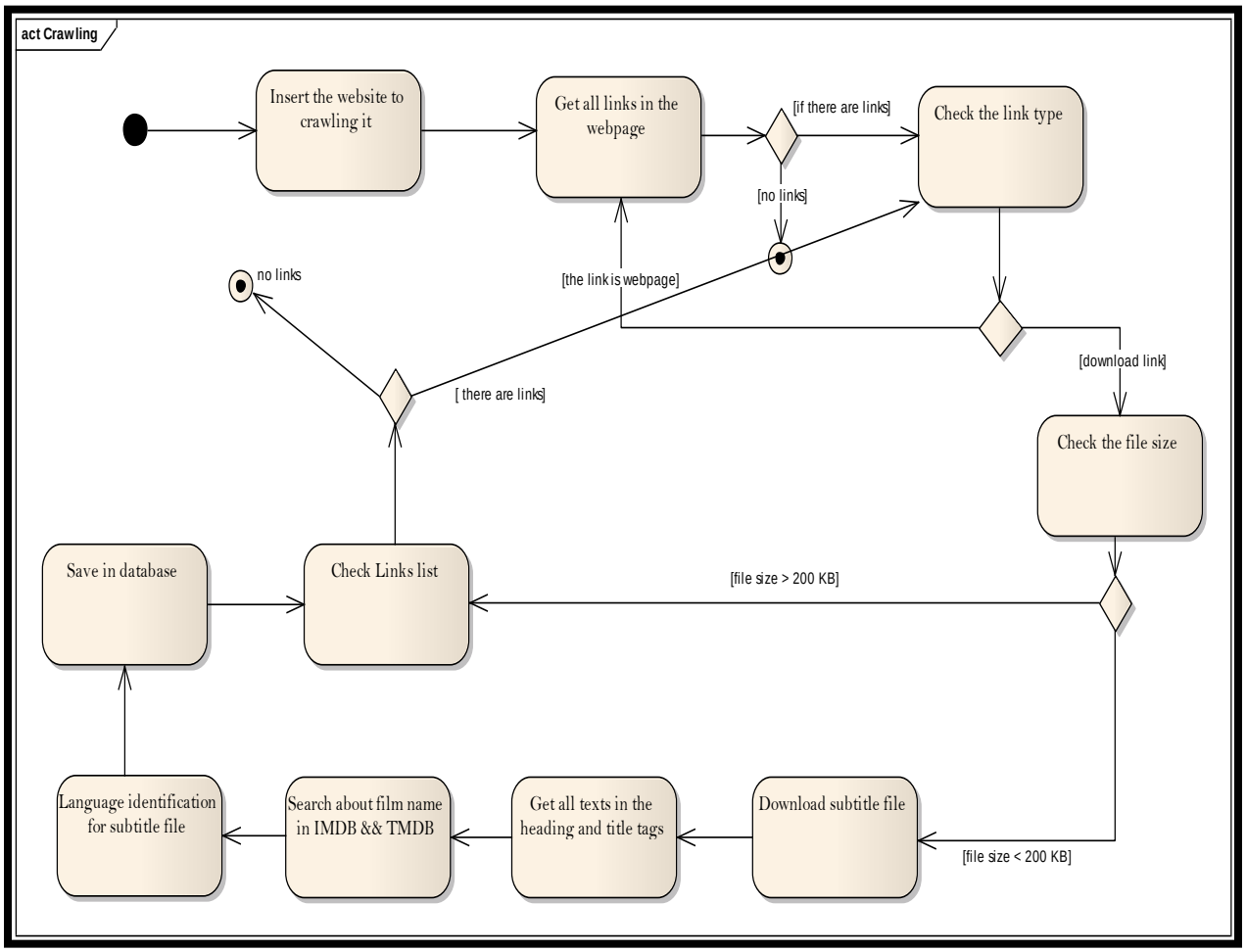
AddMovie –



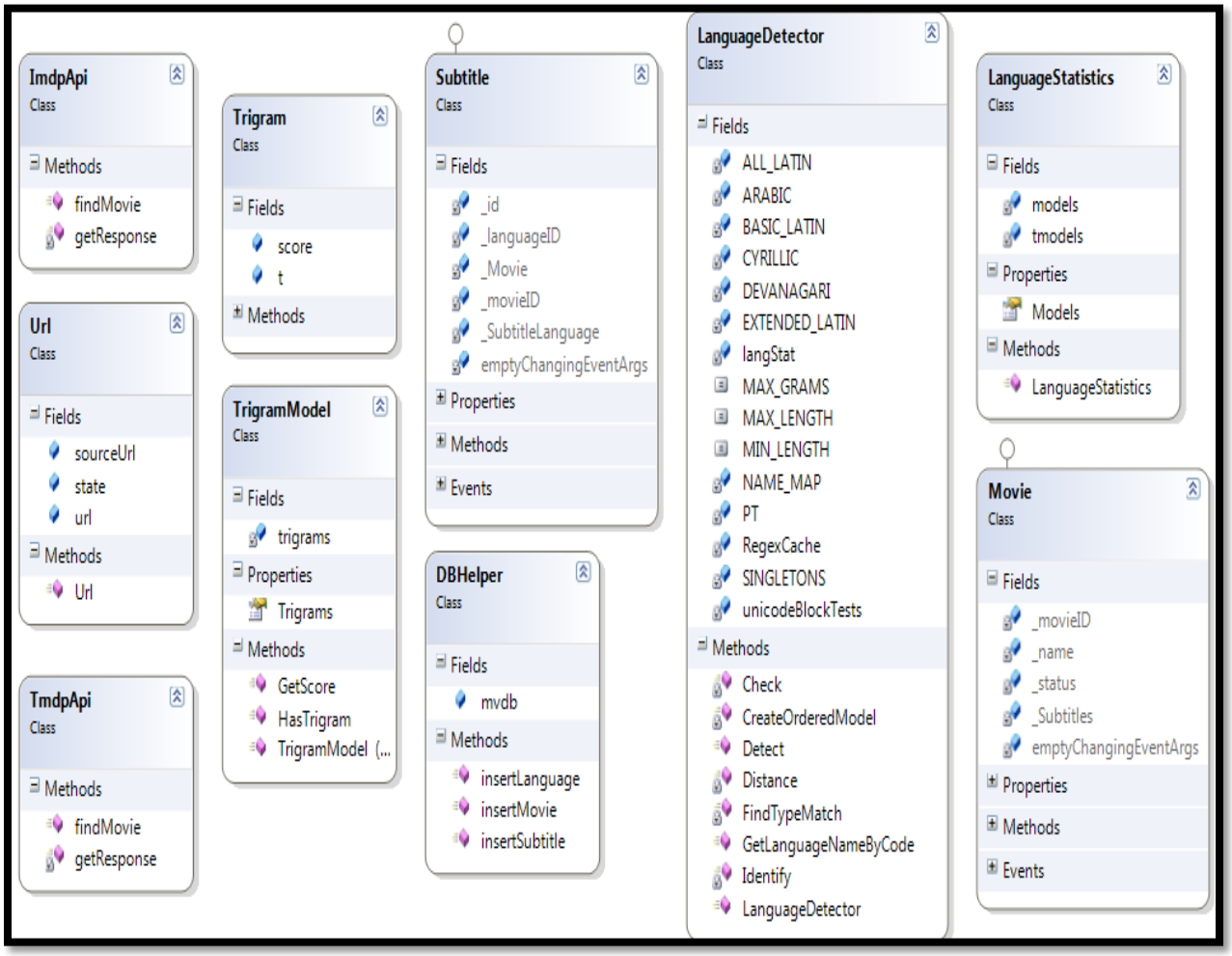
AddSubtilte –



Crawling –


act Crawling
Insert the website to crawling it
Get all links in the webpage
[if there are links]
Check the link type
[no links]
no links
[the link is webpage]
[download link]
[there are links]
Check the file size
Save in database
Check Links list
[file size > 200 KB]
Language identification for subtitle file
Search about film name in IMDB && TMDB
Get all texts in the heading and title tags
Download subtitle file
[file size < 200 KB]

4.5 مخطط الصفوف الخاص بمرحلة بناء الوكيل البرمجي



4.6 الأدوات البرمجية المستخدمة في تحقيق الوكيل البرمجي

لغة البرمجة المستخدم في تحقيق الوكيل البرمجي هي C#، واعتمادا على نظام إدارة قواعد المعطيات SQl server؛

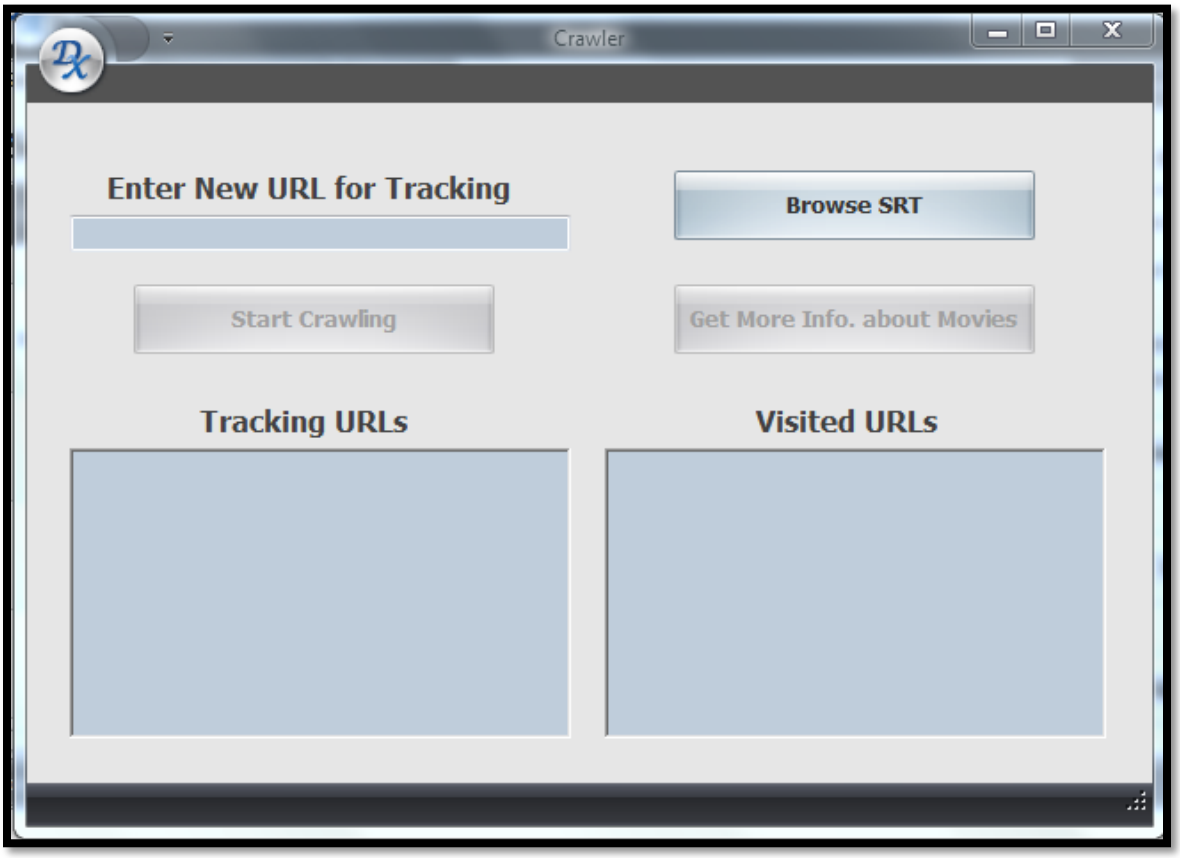
IMDB API و TMDB API تم استخدامهما للحصول على معلومات عن الأفلام، ولهما web service متاحة على الأنترنت، وهما قاعدتي معطيات ضخمة عن الأفلام تمكن المستخدمين من التعامل معها وإجراء عمليات عليها؛

مكتبة الضغط وفك الضغط Ionic.Zip والتي تم استخدامها من أجل فك ضغط الملفات المضغوطة بعد تحميلها؛

استخلاص النصوص من صفحات الموقع من خلال المكتبة HtmlAgilityPack

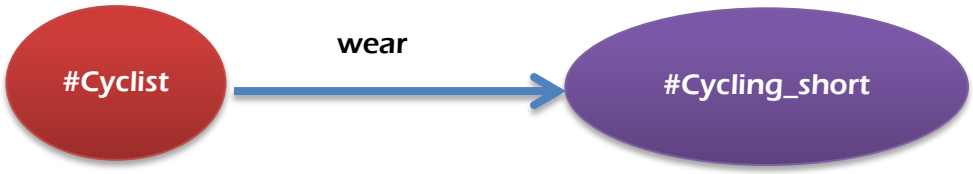
المكتبة تسمح بالتعامل مع صفحات HTML، حيث أنها تقوم بتحويل صفحة HTML إلى مجموعة من العقد والتي تمكن من البحث واستخراج المعلومات بسهولة كما أنها تتعامل حتى مع صفحات HTML سيئة التنسيق وتقوم بإصلاحها، بالإضافة أنها تدعم استخدام تعابير XPATH و XSLT وهي تعابير تستخدم للبحث داخل مجموعة من العقد واستخراج العقد التي نريد بسرعة، كما تستطيع إنشاء صفحات أو إضافة عقد إليها أو تغيير عقد موجودة، ويمكن أيضا أن تحول صفحة HTML إلى صفحة XML بأمر واحد، والكثير من المزايا الأخرى.

4.7 واجهة الوكيل البرمجي ضمن التطبيق



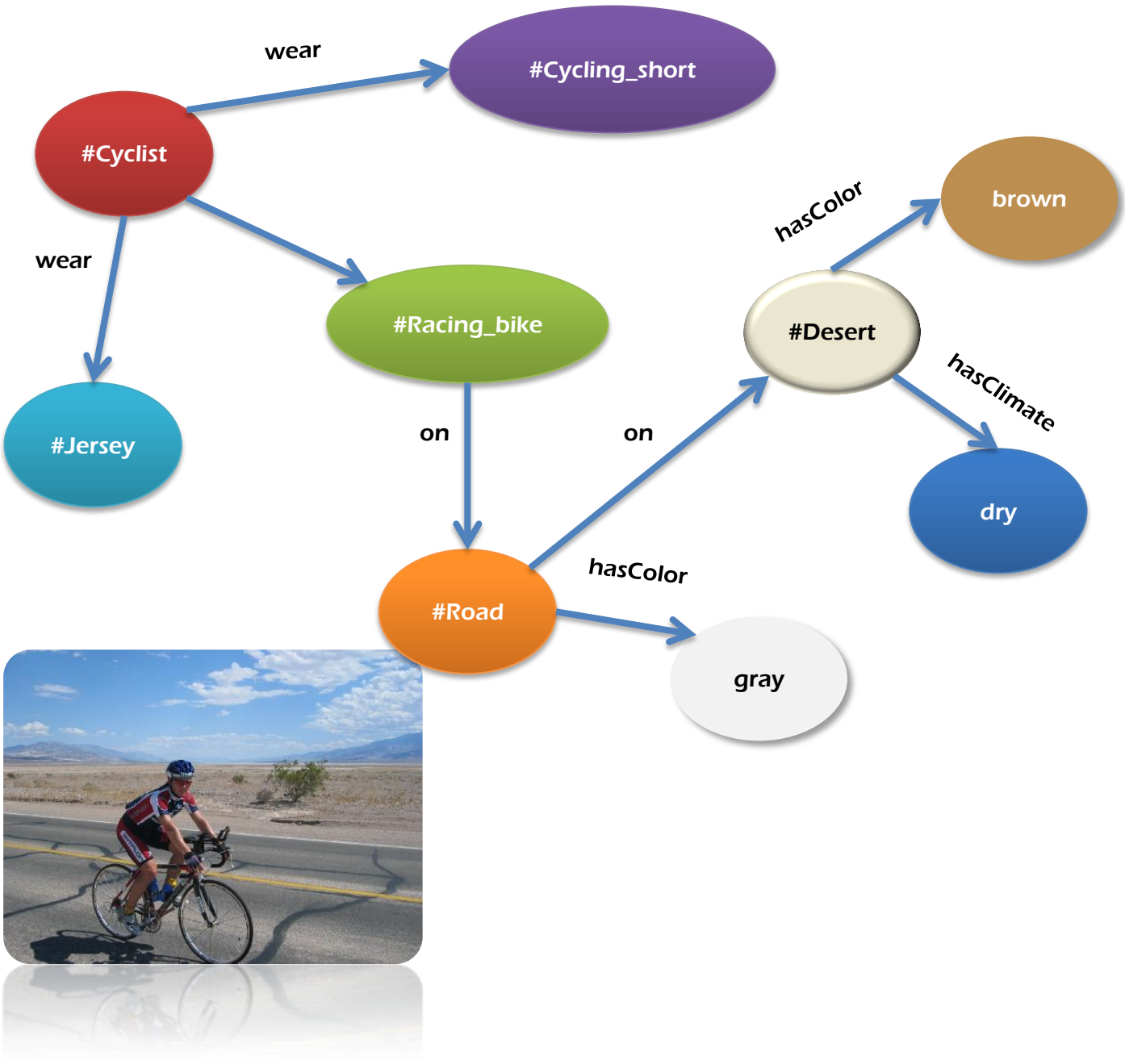
5 RDF Resource Description framework

- هو اختصار لـ (Resource Description Framework) وهي عبارة عن مجموعة من المواصفات التي وضعها اتحاد الشبكة العنكبوتية (W3C) لتعريف بنية تحتية مرنة خاصة بتنظيم وإدراة خصائص البيانات التي تسمى (metadata) في الشبكة العنكبوتية؛
- metadata هي عبارة عن معلومات وصفية عن طبيعة البيانات والوثائق مثل مصدرها، وحجمها و التنسيق الخاص بها وخصائص أخرى خاصة بالبيانات؛
- إذاً فـ RDF قد تم تصميمها لتوفير إطار يعتمد على لغة XML التي يمكنها أن تقوم بتوحيد عملية تبادل خصائص البيانات بين التطبيقات المختلفة أو ما يسمى بخصائص المحتويات (metacontent)؛
- ثلاثيات ال RDF تتألف من Subject, Predicate and Object؛



- ومن الاستخدمات المحتملة للـ RDF محركات البحث، وأنظمة تقييم المحتويات، ومجالات أخرى تهتم بخصائص البيانات المتبادلة؛

- مثال يوضح ثلاثيات ال RDF.



6 استخراج الثلاثيات RDF Triples

6.1 قراءة ومعالجة نص الترجمة

بما أن نصوص الترجمة تكون ذات لاحقة srt. فلقد استعنا بمحلل لغوي (Parser) خاص لمعالجة هذه الأنواع من النصوص يُدعى ب JSRT. إن هذا المحلل يوفر لنا آلية لقراءة نص الترجمة والتعرف على أجزاءه بالإضافة إلى أنه يقوم بالتحقق من صلاحيته (باعتبار نص الترجمة له هيكلية معينة) واكتشاف الأخطاء فيه، بالإضافة لإمكانية تعديل نص الترجمة، وإنشاء نص ترجمة جديد، بالإضافة إلى العديد من الميزات الأخرى.

لكننا اكتفينا بالاستفادة من JSRT بالعمليات المتعلقة بقراءة نص الترجمة والتحقق من صلاحيته وقمنا بتضمين أجزاء المحلل فقط التي ستفيدنا في إجراء هذه العمليات.

6.1.1 أجزاء نص الترجمة

كما نعلم أن أي نص ترجمة يتألف من عدة مشاهد حيث أن كل مشهد يتكون من الواصفات التالية:

- رقم المشهد
- زمن بدء المشهد في الفيديو
- زمن انتهاء المشهد في الفيديو
- نص المشهد (والذي من الممكن أن يمتد على أكثر من سطر)

إن المحلل JSRT يقوم بقراءة نص الترجمة وتحليل مشاهده وتخزينها في بنية خاصة تدعى SRTInfo والتي هي عبارة عن TreeSet من البنية SRT التي توصّف المشهد الواحد.

يحوي أيضاً JSRT على طرق لمعالجة زمن المشهد حيث يوفر عدة Formats لذلك.

6.1.2 نصوص الترجمة الخاطئة المكتشفة في JSRT

لقد وجدنا أنه من الفائدة اكتشاف أن نص الترجمة صالح أم لا، وذلك لأنه بعد عملية جلب نصوص الترجمة (Crawling) من الممكن أن يتم جلب أكثر من نص ترجمة لنفس الفيلم وبنفس اللغة، وبالتالي هنا نقوم بتمرير هذه الملفات الواحد تلو الأخر على المحلل فإذا اكتشفنا خطأ في أحدها ننتقل للنص الآخر وهكذا...

ومن أنواع الأخطاء الشائعة في نصوص الترجمة:

1- النصوص التي تحوي على أكثر من سطر فارغ بين كل مشهد وأخر

```
1

00:00:20,000 --> 00:00:24,400

Hello World

Bye World



2

00:00:24,600 --> 00:00:27,800

Boo Foo
```

2- النصوص التي ينقصها زمن بداية المشهد أو زمن النهاية

```
1

00:00:20,000 --> 00:00:24,400

Hello World

Bye World


2

foo boo
```

3- النصوص التي تحوي طريقة خاطئة في كتابة زمن بداية أو نهاية المشهد

```
1

00:00:20,000 --> 00:00:24,400

Hello World

Bye World


2

00:00:24,600 00:00:27,800
```

4- النصوص التي لا تحوي نص ترجمة في المشهد

1

00:00:20,000 --> 00:00:24,400

Hello World

Bye World

2

00:00:24,600 --> 00:00:27,800

3

00:00:20,000 --> 00:00:24,400

Foo Bar

6.2 التحسينات التي قمنا بإجراها على النص قبل تحويله الى ثلاثيات

قبل تطبيق عمليات تحويل النص إلى ثلاثيات قمنا بإجراء بعض عمليات المعالجة عليه، وذلك للتحسين من دقة النتائج ومن هذه العمليات:

6.2.1 استبدال الضمائر بالكلمات الأصلية التي تعود عليها هذه الضمائر Coreferencing

أحد المشاكل التي واجهتنا أثناء تحويل النص الى ثلاثيات، هي ورود اسم شخص مثلا في بداية الجملة، ثم عند ذكر أفعال أخرى قام بها نفس الشخص فإنه يتم ذكر ضمير عائد على اسم الشخص بدلا من ذكر اسمه، و هذا يمنعنا من معرفة الفاعل الحقيقي لهذا الفعل مثلا:

John played football, and **he** also played golf.

نلاحظ أن الضمير he يعود على الفاعل John، لذا نقوم باستبدال الضمير بالفاعل قبل البدء بمعالجة النص وتحويله الى ثلاثيات فيصبح :

John played football, and **John** also played golf.

قمنا بالاستعانة بخدمة **Coreference** الموجودة ضمن مكتبة **Stanford CoreNLP**، لكننا قمنا بتطبيق بعض التحسينات عليها لمعالجة بعض المشاكل مثلا اذا أدخلنا المثال التالي:

Damascus is **the capital of Syria**.

ستكون النتيجة أن جملة the capital of Syria كلها تعود على Damascus، أي ستصبح الجملة بعد تطبيق العملية عليها:

Damascus is **Damascus**.

وهذه النتيجة غير مجدية أبدا، لذا قمنا بحصر هذه العملية فقط على الضمائر بكافة أنواعها (he, she, it, they...)، وإضافة 's في حال كان الضمير هو أحد ضمائر الملكية مثل(his, her ...).

كمثال على ذلك الجملة التالية التي تملك ضمير الملكية "his":

John likes all people. So people like **his** personality.

تصبح بعد المعالجة:

John likes all people. So people like **John's** personality.

6.2.2 إعادة الكلمات إلى أصلها Lemmatization

لمزيد من التحسين على النتائج قمنا أيضا بإعادة الكلمات إلى أصلها، حيث قمنا في البداية بتطبيق التجذير stemming على الكلمات لكن استنتجنا أن هذه الطريقة تحوي على العديد من المشاكل، فمثلا كلمة providers تتحول بتطبيق خوارزمية porter –وهي أحد خوارزميات التجذير– إلى كلمة provid

حيث نلاحظ أنها قامت بإزالة حرف ال e أيضا من نهاية الكلمة و بالتالي لم تعد الكلمة صحيحة لغويا ولا توجد في أي معجم.

لذلك قمنا باستخدام عملية أخرى تدعى lemmatization حيث تقوم هذه الطريقة بإرجاع الكلمة إلى أصلها لكن بالاستعانة بمعجم مثل WordNet وبالتالي نتائجها تكون صحيحة و موجودة في المعجم.

مثلا كلمة providers بعد تطبيق العملية عليها تعود الى كلمة provider.

ولتطبيق هذه العملية قمنا بالاستعانة بخدمة موجودة ضمن مكتبة **Stanford CoreNLP**.

6.2.3 تحويل الجمل المعقدة و الجمل المكربة الى جمل بسيطة

الهدف من هذه العملية هو تحويل الجملة الطويلة والمعقدة إلى جمل بسيطة ومفهومة بحيث يسهّل استخراج الثلاثيات منها، كمثال عن ذلك تبسيط الجملة التالية:

John, who was the CEO of a company, played golf.

إلى مجموعة الجمل التالية:

John played golf.

John was the CEO of a company.

لإجراء هذه العملية نعتمد على مكتبة Stanford في إيجاد قواعد Typed Dependencies للجملة الأصلية، حيث يتم تمثيل الجملة هنا بمجموعة من العلاقات النحوية حيث أن كل علاقة تكون بين زوج من الكلمات، وإن هذه الطريقة في تمثيل الجملة تعطي توصيف بسيط للعلاقات النحوية بين كلمات الجملة بحيث تُمكن المستخدمين الذين يريدون استخراج هذه العلاقات من فهمها بسهولة دون الحاجة لامتلاك الخبرة.

المرجع (2) يوضح هذه العلاقات مع أمثلة عن كل منها

بتطبيق قواعد Typed Dependencies على الجملة الأصلية في مثالنا نحصل على:

nsubj(CEO-6,John-1)
nsubj(played-11,John-1)
cop(CEO-6,was-4)
det(CEO-6,the-5)
rcmod(John-1,CEO-6)
det(company-9,a-8)

prep_of(CEO-6,company-9)

root(ROOT-0,played-11)

dobj(played-11, golf-12)

نأخذ العلاقات بين الزوج (subject, verb) والتي هي هنا (3):

nsubj(CEO-6,John-1)

nsubj(played-11, John-1)

سنبدأ المعالجة على العلاقة الأولى nsubj(CEO-6, John-1)

نبحث عن جميع العلاقات الجديدة التي تحوي إحدى الكلمتين CEO, John ما عدا العلاقات التي هي من النوع Subject، فإذا بدئنا بالكلمة John نلاحظ أنه لا توجد أي علاقة تحقق المطلوب، بالانتقال للكلمة الثانية CEO نجد العلاقات التالية التي تحقق المطلوب:

cop(CEO-6,was-4)

det(CEO-6,the-5)

rcmod(John-1,CEO-6)

prep_of(CEO-6, company-9)

نكرر نفس العملية على مجموعة العلاقات التي حصلنا عليها، أي نبحث عن جميع العلاقات الجديدة التي تحوي على إحدى الكلمات التالية company, the, was فنجد العلاقة

det(company-9, a-8)

نلاحظ هنا أنه لم نعد نستطيع الحصول على علاقات جديدة من العلاقات التي لدينا، فنتوقف هنا وتكون جملة العلاقات النهائية التي حصلنا عليها هي:

cop(CEO-6,was-4)

det(CEO-6,the-5)

rcmod(John-1,CEO-6)

prep_of(CEO-6,company-9)

det(company-9, a-8)

الآن يتم المرور على العلاقات الناتجة وترتيب الكلمات الموجودة فيها اعتماداً على الرقم المذكور بعد كل كلمة والذي يعبر عن رقم هذه الكلمة ضمن الجملة لنحصل على الجملة التالية:

John was the CEO a company

بتطبيق نفس الطريقة على العلاقة الثانية:

nsubj(played-11, John-1)

نحصل على الجملة الثانية

John played golf

وبذلك نلاحظ كيف تم تقسيم الجملة المعقدة إلى مجموعة جمل بسيطة ومفهومة وسهلة لاستخراج الثلاثيات منه.

بعد إجراء التجارب على مجموعة من الجمل وتطبيق طريقة التبسيط التي ذكرناها وجدنا أن هذه الطريقة غير فعالة كثيراً في عملية التحسين لذلك لم نجد جدوى في برمجتها.

6.2.4 حذف المحارف و الرموز الغريبة

لمزيد من التحسين على النتائج، قمنا بإضافة تابع آخر لمعالجة النص قبل استخراج الثلاثيات منه، حيث يقوم هذا التابع بحذف أية رموز غريبة، أو أية محارف تنتمي إلى لغات أخرى غير اللغة الإنكليزية.

6.3 تحويل النص إلى ثلاثيات Offline (Method1)

الدخل: نص الترجمة بعد معالجته.

الخرج: الثلاثيات الناتجة من النص.

خطوات الخوارزمية:

1. تقطيع النص إلى جمل حيث يتم تمييز الجمل عن بعضها من خلال المحرف "." ؛
2. من أجل كل جملة يتم المرور بالخطوات التالية:

2.1. تقطيع الجملة إلى مجموعة الكلمات Tokens التي تحويها؛

2.2. تطبيق CoReference؛

2.3. تطبيق Lemmatization؛

2.4. إيجاد الشجرة النحوية (Parse Tree) للجملة الحالية حيث أن رأس الشجرة هو العقدة S أي Sentence وأولادها المباشرين هم: NP (Noun Phrase), VP (Verbal Phrase), . (full stop).

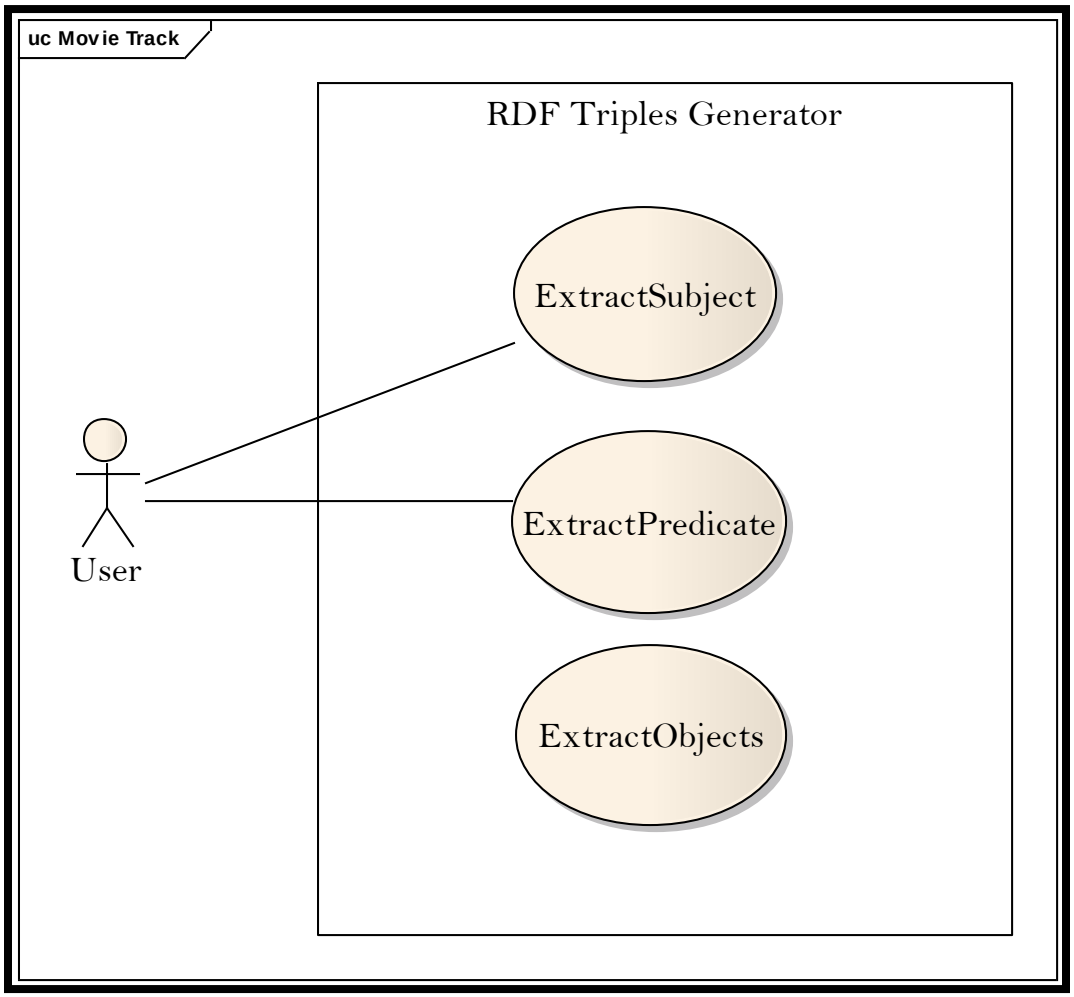
2.5. استخراج الثلاثيات الممكنة من الجملة: (1)

2.5.1. استخراج Subject

2.5.2. استخراج Predicate

2.5.3. استخراج Objects

6.4 حالات الاستخدام المستخدمة في استخراج الثلاثيات



حالة الاستخدام	ExtractSubject
رقم حالة الاستخدام	10
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	استخراج ال Subject من جملة محددة.
الشروط المسبقة	تحديد الجمل وتحسينها وتقسيم النص المدخل إلى tokens.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام استخراج ال Subject من جملة ما؛ 2- تقطيع النص إلى جمل حيث يتم تمييز الجمل عن بعضها من خلال المحرف "." ؛ 3- من أجل كل جملة يتم المرور بالخطوات التالية: a. تقطيع الجملة إلى مجموعة الكلمات Tokens التي تحويها؛ b. تطبيق CoReference؛ c. تطبيق Lemmatization؛ d. إيجاد الشجرة النحوية (Parse Tree) للجملة الحالية حيث أن رأس الشجرة هو العقدة S أي Sentence وأولادها المباشرين هم: NP (Noun Phrase), VP (Verbal Phrase), . (full stop). 4- الحصول على شجرة ال NP (أحد أبناء رأس الشجرة S)؛

	5- تطبيق البحث بالعرض (Breadth First Search) عن أول اسم نجده في شجرة ال NP والذي يكون في الشجرات الجزئية التالية (NN, NNP, NNPS, NNS)، والذي يمثل Subject في هذه الجملة.
الشروط اللاحقة	تم إيجاد Subject الجملة.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

حالة الاستخدام	ExtractPredicate
رقم حالة الاستخدام	11
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	استخراج ال Predicate من جملة محددة.
الشروط المسبقة	تحديد الجمل وتحسينها وتقسيم النص المدخل إلى tokens.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام استخراج ال predicate من جملة ما؛ 2- تقطيع النص إلى جمل حيث يتم تمييز الجمل عن بعضها من خلال المحرف "." ؛ 3- من أجل كل جملة يتم المرور بالخطوات التالية: a. تقطيع الجملة إلى مجموعة الكلمات Tokens التي تحويها؛

	b. تطبيق CoReference؛ c. تطبيق Lemmatization؛ d. إيجاد الشجرة النحوية (Parse Tree) للجملة الحالية حيث أن رأس الشجرة هو العقدة S أي Sentence وأولادها المباشرين هم: NP (Noun Phrase), VP (Verbal Phrase), . (full stop). 4- إيجاد شجرة ال VP (أحد أبناء رأس الشجرة S)؛ 5- تطبيق البحث بالعمق (Depth First Search) عن أبعد فعل نجده في شجرة ال VP عن رأس الشجرة (أي الأعمق) والذي يكون في الشجرات الجزئية التالية (VB, VBD, VBG, VBN, VBP, VBZ)، والذي يمثل Predicate في هذه الجملة.
الشروط اللاحقة	تم إيجاد Predicate الجملة.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

حالة الاستخدام	ExtractObjects
رقم حالة الاستخدام	12
الممثلين الأوليين	مستخدم النظام
الممثلين الثانويين	لايوجد.
توصيف مختصر	استخراج ال Objects من جملة محددة.
الشروط المسبقة	تحديد الجمل وتحسينها وتقسيم النص المدخل إلى tokens وتحديد ال Predicate.
التدفق الأساسي للأحداث	1- تبدأ حالة الاستخدام هذه عندما يريد مستخدم النظام استخراج ال Objects من جملة ما؛ 2- تقطيع النص إلى جمل حيث يتم تمييز الجمل عن بعضها من خلال المحرف "." ؛ 3- من أجل كل جملة يتم المرور بالخطوات التالية: a. تقطيع الجملة إلى مجموعة الكلمات Tokens التي تحويها؛ b. تطبيق CoReference؛ c. تطبيق Lemmatization؛ d. إيجاد الشجرة النحوية (Parse Tree) للجملة الحالية حيث أن رأس الشجرة هو العقدة S أي Sentence وأولادها المباشرين هم: NP (Noun Phrase), VP (Verbal Phrase), . (full stop).

	4- إيجاد الأشجار الجزئية الشقيقة مع ابن شجرة ال VP التي تم العثور فيها على Predicate، 5- من أجل هذه الأشجار الجزئية الشقيقة: a. إذا كانت قيمة رأس الشجرة هو PP أو NP يتم البحث بالعرض عن اول اسم نجده ليكون هو Object الثلاثية. b. وإلا إذا كانت قيمة رأس الشجرة هو ADJP يتم البحث بالعرض عن اول صفة (Adjective) والتي يمكن أن تكون في الشجرات الجزئية التالية (JJ, JJR, JJS) لتكون هي ال Object في الثلاثية.
الشروط اللاحقة	تم إيجاد Objects الجملة.
الطرق البديلة	لايوجد.
لاستثناءات	لايوجد.

ملاحظة

نلاحظ أنه من الممكن أن يتم العثور على أكثر من Object لنفس ال Subject وال Predicate، وبالتالي من الممكن أن تتشكل أكثر من ثلاثية في الجملة الواحدة.

مثال عن الخوارزمية السابقة

إذا كان لدينا النص التالي كدخل:

A rare black squirrel has become a regular visitor to a suburban garden. This garden locates in NewYork city.

ستكون الثلاثيات الناتجة:

(squirrel, become, visitor)

(garden, locates, NewYork)

حيث سيتم أولا تقسيم النص إلى مجموعة جمل هي:

A rare black squirrel has become a regular visitor to a suburban garden.

This garden locates in NewYork city.

سيتم معالجة كل جملة فمثلاً إذا اخذنا الجملة الأولى:

تقسيمها إلى مجموعة الكلمات Tokens التي تحويها:

[A, rare, black, squirrel, has, become, a, regular, visitor, to, a, suburban, garden, .]

نلاحظ أنه لا يوجد في الجملة أي ضمير عائد وبالتالي يتم تجاهل خطوة تطبيق CoReferencing.

يتم إيجاد الشجرة النحوية واستخراج الثلاثية باعتماد الخطوات التي ذكرناها سابقاً والشكل التالي يوضح الشجرة والثلاثية الناتجة (1):

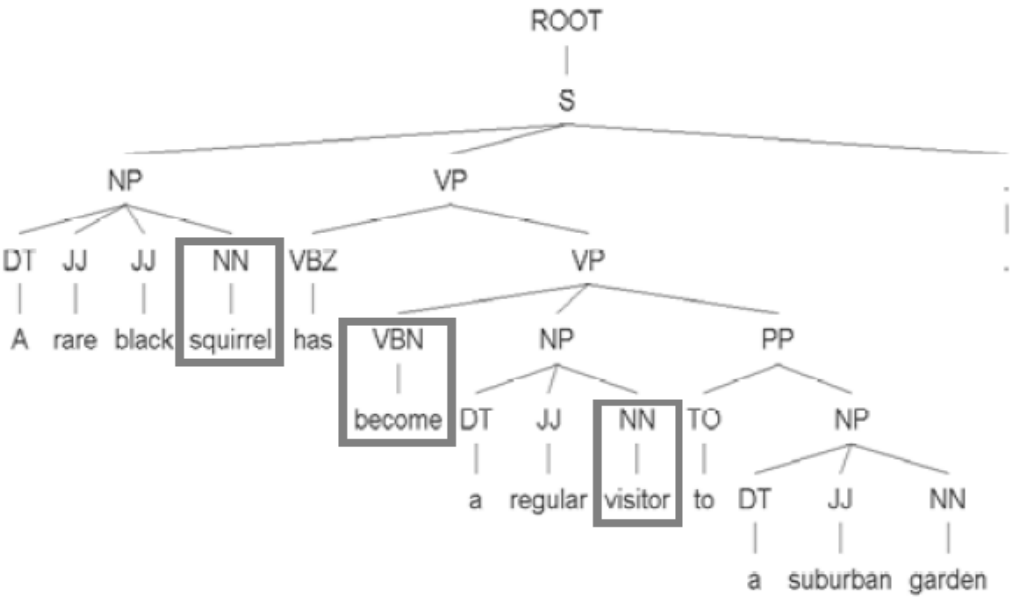
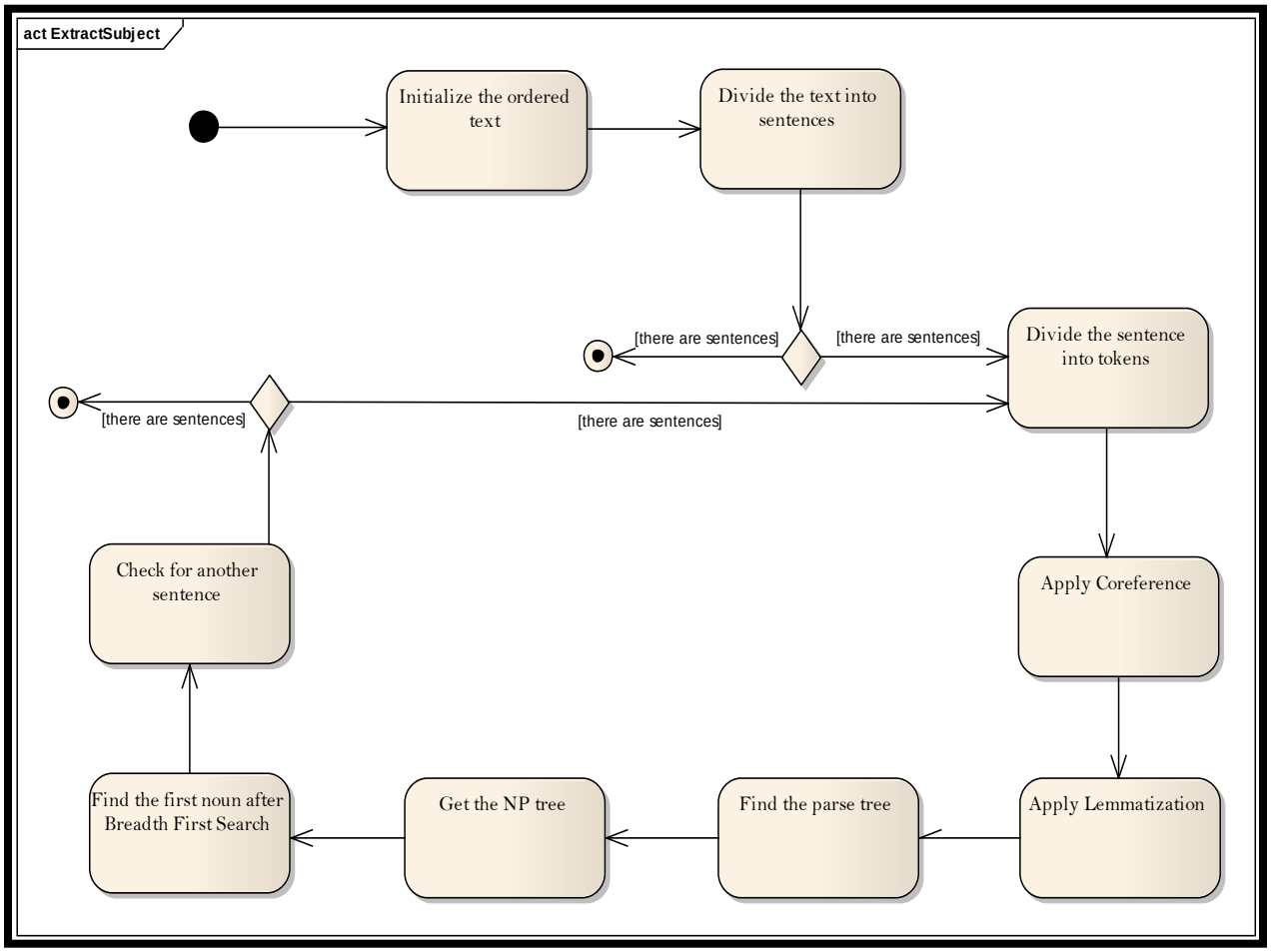


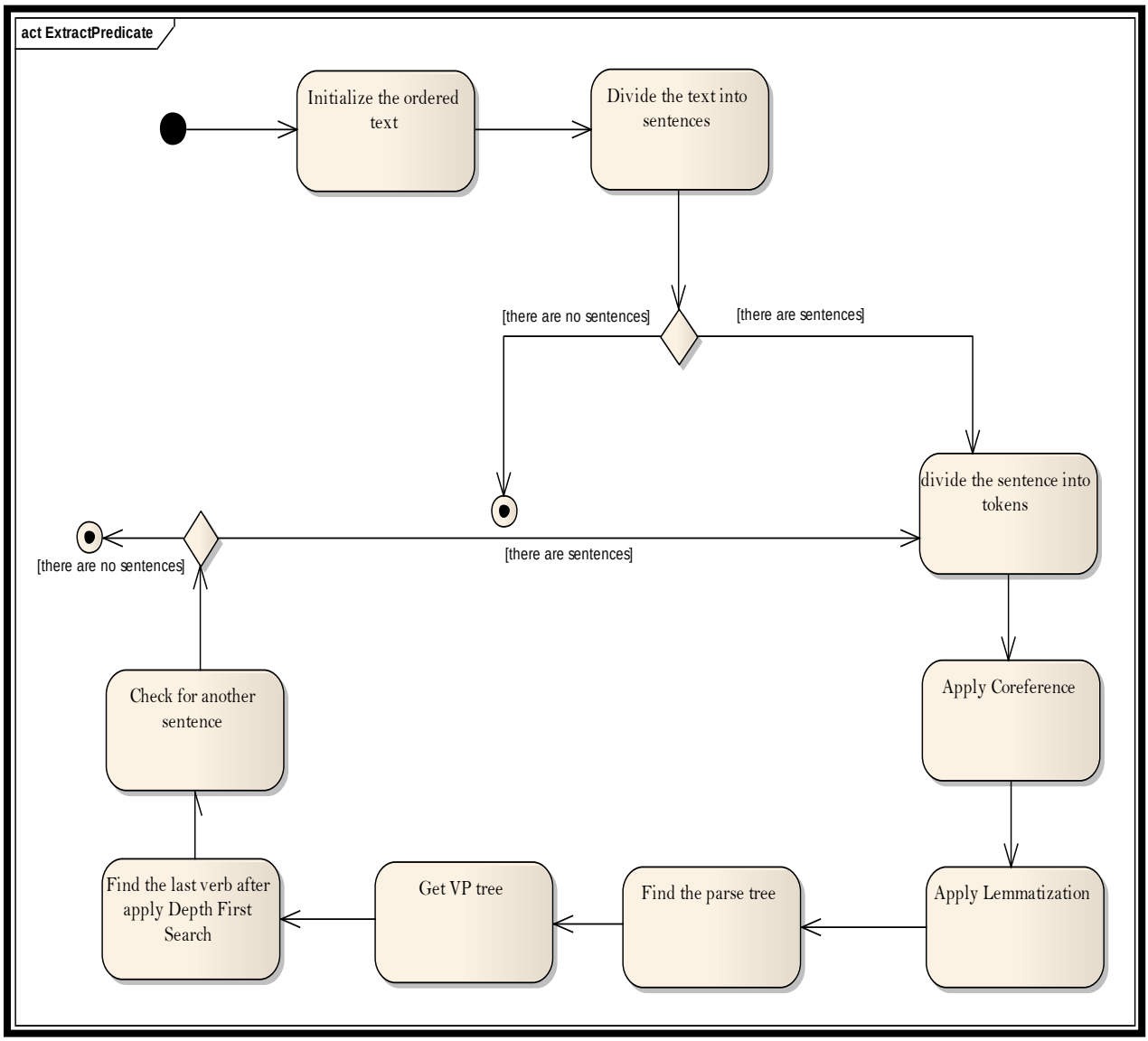
Figure 2: *The parse tree generated by Stanford Parser, when the input sentence is "A rare black squirrel has become a regular visitor to a suburban garden." The triplet extracted out of this sentence is* ***squirrel – become – visitor.***

6.5 مخططات الأنشطة Activity Diagrams الخاصة بمرحلة توليد الثلاثيات

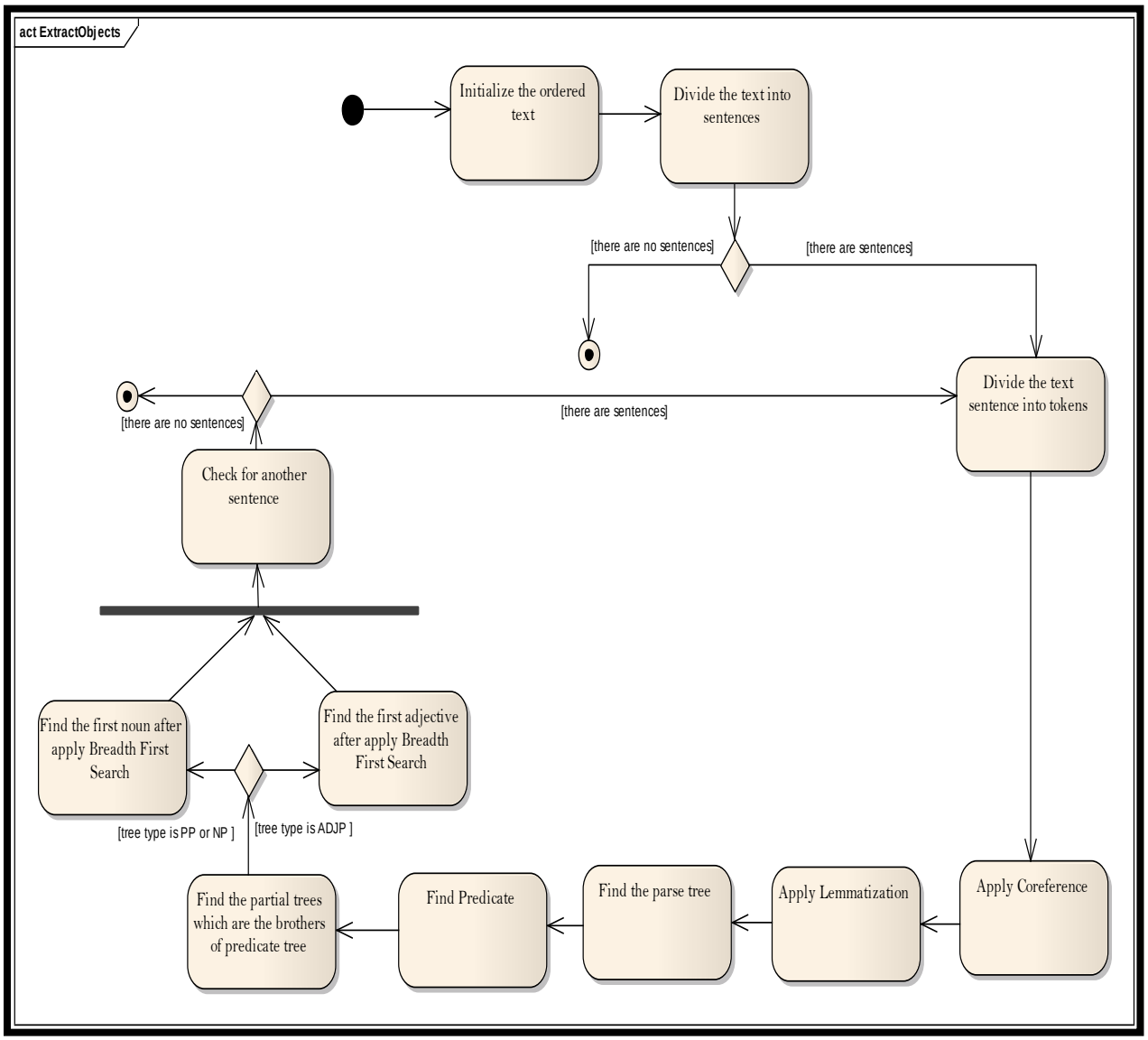
ExtractSubject –



ExtractPredicate –



ExtractObjects –



6.6 تحويل النص إلى ثلاثيات Offline (Method2)

لقد قمنا بإتاحة طريقة أخرى أيضا لتحويل النص إلى ثلاثيات و ذلك بالاعتماد على قواعد type dependency بالاستعانة بالمكتبات التالية:

```
Entities.Sentence;
Entities.Triple;
edu.stanford.nlp.objectbank.TokenizerFactory;
edu.stanford.nlp.process.CoreLabelTokenFactory;
edu.stanford.nlp.process.PTBTokenizer;
edu.stanford.nlp.ling.CoreLabel;
edu.stanford.nlp.trees.*.
```

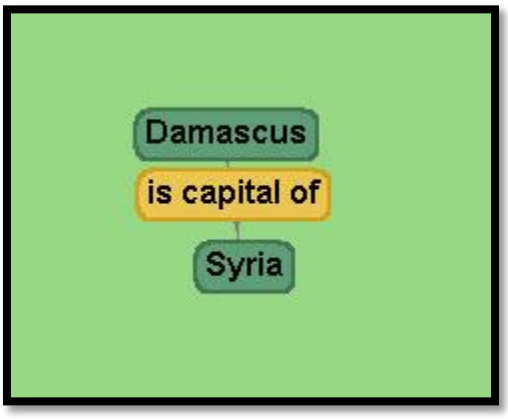
6.7 تحويل النص إلى ثلاثيات بالاستعانة ب Web Service (Method3)

لقد قمنا بإتاحة طريقة أخرى أيضا لتحويل النص إلى ثلاثيات و ذلك بالاستعانة بخدمة ويب خارجية تدعى Enrycher، حيث أن دخل هذه الخدمة هو نص و خرجها ملف XML يحوي العديد من المعلومات عن النص منها مجموعة الثلاثيات المستخرجة من النص.

فإذا أخذنا هذا النص كمثال مثلا:

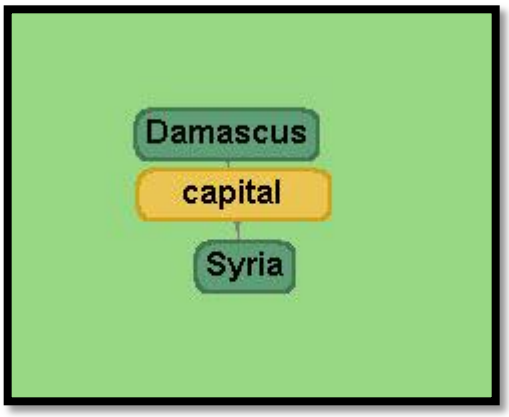
Damascus is the capital of Syria.

ستقوم الخدمة باستخراج هذه الثلاثية من النص



و كما نلاحظ أنه يوجد مشكلة هنا وهي أن ال predicate هنا ليست عبارة عن كلمة واحدة، لذلك قمنا بإجراء تحسين على نتائج هذه الخدمة لتحويل ال predicate إلى كلمة واحدة، وذلك بعد تجربة عدد كبير من الأمثلة، حيث قمنا بحذف أحرف الجر من آخر ال predicate و من ثم الاحتفاظ بآخر كلمة كقيمة جديدة لل predicate.

بعد تطبيق عمليه التحسين تصبح الثلاثية على الشكل التالي:



ومن ميزات هذه الطريقة هو الحصول على predicate حتى لو لم يكن هناك أفعال ضمن الجملة،فعلى سبيل المثال نلاحظ في الجملة التالية عدم وجود فعل واضح ضمن الجملة

John is the CEO of company.

عند استخراج الثلاثيات نحصل على:

Subject = John

Predicate = CEO

Object = company

وفي الجملة التالية أيضا نلاحظ عدم وجود فعل

Damascus is the capital of Syria

عند استخراج الثلاثيات نحصل على:

Damascus Subject = Damascus

Predicate = Capital

Object = Syria

6.8 تحويل النص إلى ثلاثيات Offline (Method4)

تشمل هذه الطريقة حاصل اجتماع الطريقتين Method1 و Method2 بهدف الحصول على أكبر عدد ممكن من الثلاثيات بشكل Offline دون الاتصال بالإنترنت.

Method4 = Method1 + Method2

6.9 تحويل النص إلى ثلاثيات Online (Method5)

تشمل هذه الطريقة حاصل اجتماع الطرق Method1, Method2 and Method3 بهدف الحصول على أكبر عدد ممكن من الثلاثيات بشكل Online نظرا لاستخدام Method3.

Method5 = Method1 + Method2 + Method3

Wordnet Domains 6.10

- Wordnet domains هو تصنيف لكل synset ضمن ال wordnet بالنسبة لكل ال domains المتاحة وكم نسبة انتماء هذه ال synset إلى كل domain؛
- يوجد حوالي 170 domain؛
- هناك 5 abstract domains ويتفرع عنها باقي ال domains؛

- تم استخدام wordnet domains لتحديد ال domain الخاص بكل ثلاثية وتخزينه ضمن ملف ال RDF وبالتالي فلترة نتائج البحث بناء على ال domain المحدد في الاستعلام.

SpotLight 6.11

- Spotlight هي أداة للتأشير التلقائي للنصوص على موارد ال DBpedia؛
- توفير حل لربط مصادر المعلومات غير المهيكلة لفتح سحابة البيانات المرتبطة من خلال DBpedia؛

- توفر web service دخلها النص وخرجها ال annotated text؛
- يمكن تمرير النص كامل وتحديد كلمات محددة للتأشير عليها وهذا ما نفعله نحن في تطبيقنا، حيث ما نريد التأشير عليه هو ال Subject وال Predicate؛
- قمنا بتمرير الجملة كاملة في محاولة لإزالة غموض الكلمات، حيث تقدم ال spotlight أيضا خدمة فك غموض النص الممرر لها.

6.12 تحسينات تم تطبيقها على ملفات ال RDF المولدة للحصول على نتائج أفضل عند البحث

- من أجل كل ثلاثية يتم الحصول على ال domain الخاص بها، حيث تم الاستفادة من خوارزمية يتم العمل عليها ضمن مشروع التخرج الخاص بنا، حيث يتم تمرير هذه الثلاثية كما لوكانت نص عادي، ليتم إجراء عمليات تحليل نصية على هذه الثلاثية والحصول على أكثر domain تنتمي له هذه الثلاثية، وتخزين ال domain ضمن ملف ال RDF، ليتم الاستفادة منه لاحقا في عمليات الاسترجاع، فمن الممكن أن يحدد المستخدم domain محدد من أجل استعلامه وبالتالي لن يتم إحضار إلا الثلاثيات المنتمية إلى هذا ال domain والموافقة للاستعلام المطلوب، ولتحقيق المطلوب يتم المرور بالمراحل التالية:

1- تجزئة الكلمات التي تحوي على أحرف كبيرة في منتصفها أو التي تحوي المحرف "-" فذلك يعني أنها كلمة جاءت كتركيب أو أنها اسم علم؛

2- إزالة المحارف الغريبة من النص مثل المحارف التي لا تنتمي إلى الأبجدية الإنكليزية أو الأرقام لأن لا معنى لها ضمن معالجتنا للنصوص؛

3- تقسيم النص إلى جمل حسب علامات الترقيم حيث يتم تمييز جملة عن أخرى من خلال المحرف "." وتمييز الكلمات ضمن الجمل ومعرفة موقع الكلمة ضمن الجملة هل هي فعل أم اسم أم صفة أم ظرف؛

4- من أجل كل جملة من الجمل التي تم الحصول عليها من المرحلة السابقة:

4-1) من أجل كل كلمة token ضمن الجملة؛

4-1-1) يتم استبدال الضمائر بالكلمات الأصلية التي تعود عليها هذه الضمائر Coreferencing وذلك ضمن خيار يتم تفعيله أو إلغاء تفعيله من قبل مستخدم النظام كون هذه العملية تستغرق وقتا؛

4-1-2) التحقق من موقع الكمة ضمن الكلام هل هو اسم أو صفة؛

4-1-3) التحقق من أن هذه الكلمة هي كلمة تحمل معنى أي أنها لا تنتمي إلى مجموعة ال stopwords

4-1-4) التحقق من أن طول الكلمة يتجاوز المحرفين كون أن أي كلمة تتألف من محرفين فقط هي كلمة لا تحمل معنى ويجب الاستغناء عنها؛

✓ **الخوارزمية المتبعة لإزالة الغموض**

- تقوم فكرة الخوارزمية على أنه عادة ما تأتي المعاني المرتبطة بمعنى معين ضمن الكلمات المحيطة بهذا المعنى، و هذا يساعد على معرفة المعنى الحقيقي للكلمة من بين جميع المعاني المحتملة

لفرض مثلا انه لدينا الجملة التالية و لنحاول معرفة المعنى الصحيح لكلمة **pen** ضمن هذه الجملة:

.The **pen** is taking care of the small swans

- نبحث عن المعاني المحتملة لكلمة pen ضمن WordNet فنجد انها يمكن ان تأخذ المعاني التالية:

1. a writing implement with a point from which ink flows
2. an enclosure for confining livestock
3. a portable enclosure in which babies may be left to play
4. a correctional institution for those convicted of major crimes
5. female swan

- نوجد الآن المعاني المرتبطة بكل معنى من هذه المعاني بالاستعانة WordNet Semantically Tagged glosses:

1. a writing implement with a point from which ink flows
   a. point
   b. ink
   c. writing implement
2. an enclosure for confining livestock
   a. enclosure
   b. livestock, stock, farm animal
3. a portable enclosure in which babies may be left to play
   a. portable
   b. enclosure
   c. baby, babe, infant
4. a correctional institution for those convicted of major crimes
   a. major
   b. crime, offense, criminal offense, criminal offence, offence, law-breaking
   c. correctional institution
5. female swan
   a. female
   b. swan

- نقارن المعاني المرتبطة بكل معنى من المعاني الخمسة لكلمة pen مع جميع المعاني المحتملة للكلمات المحيطة بكلمة pen في الجملة المدخلة و نجمع 1 لل score الخاص بأحد المعاني الخمسة في حال حصل تطابق؛

- نلاحظ ان المعنى الخامس يأخذ ال score = 1 لتطابق المعنى swan من المعاني المرتبطة به مع احد معاني كلمة swan الموجودة في الجملة المدخلة، و المعاني الاربعة الأخرى تأخذ ال score = 0 لذلك يكون المعنى الحقيقي للكلمة هو المعنى الخامس؛
- في حال تساوى اكثر من معنى بال score وكان هذا اعلى scoreنأخذ المعنى الاكثر استخداما في اللغة (يمكننا الحصول عليه من WordNet) من بين هذين المعنيين.

4-1-5) الحصول على أصل الكلمة من خلال عملية ال Lemmatization وذلك بهدف الحصول على كلمة موجودة ضمن معجم wordnet والاستغناء بذلك عن عملية ال stemming؛

4-1-6) محاولة إزالة غموض هذه الكلمة والحصول على المعنى الحقيقي لها حسب سياق الجملة الواردة ضمنها؛

4-1-7) إيجاد المجالات التي تنتمي لها الكلمة domain من خلال الاستعانة ب wordnet domains ومن ثم الحصول على وزن هذه الكلمة ضمن كل domain والانتقال إلى شجرة المجالات وزيادة مجموع كل domain بوزن الكلمة ضمن هذا ال domain وزيادة مجموع المجالات الآباء التي يتبع لها ال domain الحالي؛

4-1-8) في حال لم تكن الكلمة موجودة ضمن معجم ال wordnet نحاول البحث عنه ضمن أنطولوجية dbpedia والاستفادة من علاقة ضمن هذه الأنطولوجية والتي تشرح معنى هذه الكلمة حيث يتم إعادة تطبيق نفس الخطوات من أجل شرح معنى هذه الكلمة ضمن dbpedia.

5- نقوم بعملية normalization للأوزان التي حصلنا عليها بعد تطبيق المراحل السابقة ضمن شجرة wordnet domains tree حيث يتم إعطاء ال domain الحاصل على أعلى وزن القيمة 1 وإعادة توزين ال domains المتبقية اعتمادا على نسبة مئوية من ال domain الأعلى وزن؛

6- اختيار ال domain الحاصل على النبسة الأعلى وإضافته إلى الثلاثية ضمن ملف ال RDF.

6.13 ربط المخزن الشخصي الذي تم إنشاؤه بأحد المخازن العامة

- تم عمل annotation لل subject وال predicate مع ما يقابلهم في المخزن العام dbpedia وذلك باستخدام spot light web service؛
- يتم تمرير الثلاثية إلى تابع مهمته عمل annotate للثلاثية الممررة، يتم استخلاص ال subject من الثلاثية مع مقدار انزياحه عن بداية الجملة، كما يتم استخلاص ال predicate من الثلاثية مع مقدار انزياحه عن بداية الجملة،
- يتم تمرير المعلومات السابقة إلى تابع getResponse والذي يقوم بالاتصال بال web service والحصول على نتيجة ال annotation؛
- في حال كانت نتيجة ال annotation تحوي عقدة واحدة يتم الحصول على URI1 مع مقدار ال Offset، وفي حال كانت نتيجة ال annotation تحوي عقدتين يتم الحصول على ال URI2 مع مقدار ال Offset، وفي النهاية يتم مقارنة انزياح ال subject مع انزياح النتيجة ومعرفة ال URI هل هو لل subject أم لل predicate؛
- في حال لم نحصل على URI لأحدهما أو لكلاهما يتم وضع dummy URI.

6.14 اقتراح آليات يدوية لتحسين صفوف البيانات وبناء نموذج أولي للواجهات الممكن استخدامها لهذا الغرض

- يتم التعديل على ملف ال RDF إما من خلال عرض قائمة Grid تحوي كل ال dummy uris الموجودة ضمن ملف ال RDF المختار واختيار المفهوم الذي يريد تعديل ال uri الخاص به، أو من خلال عرض المخطط المفاهيمي المقابل لملف ال RDF واختيار العقدة التي تريدها المستخدم وتعديل ال uri الخاص بها؛
- هذا التعديل يتم إما من خلال إدخال uri بشكل مباشر، أو من خلال عرض قائمة تضم كل ال abstract types الموجودة في dbpedia وإعطاء المستخدم الحرية في اختيار النمط الذي يراه مناسب؛
- عند إدخال uri بشكل يدوي يتم فحص هذا الرابط هل هو صحيح أم لا، هل هو موجود ضمن ال dbpedia أم لا.

6.15 إضافة DBpedia Abstract Types إلى ثلاثيات ملف ال RDF

- بعد عملية ال annotation للثلاثيات المولدة والموجودة ضمن ملف ال RDF، يتم من أجل كل triple ومن أجل كل subject ضمن هذا ال triple يتم الحصول على الأنماط المجردة **abstract types** من خلال استعلام sparql يتم على dbpedia، حيث يتم تخزين الثلاثية الأصلية مع الثلاثيات الناتجة عن تبديل كل subject بكل ال **abstract types** الموافقة له.

6.16 توليد ملفات RDF

تم توليد ملفات ال RDF انطلاقا من الثلاثيات المولدة وبالاستعانة بمكتبة ال dotNetRDF، حيث تم مقابلة كل نص ترجمة "ملف Subtitle" بملف RDF وبالتالي نستطيع الحصول على اسم الفيلم من قاعدة المعطيات اعتمادا على رقم ملف نص الترجمة Subtitle.

بالنسبة لتاريخ زمن المشهد فقد تم تخزين زمن الثلاثية من خلال إلحاق زمن الثلاثية ب Object الثلاثية المخزنة ضمن ملف ال RDF المقابل لملف نص الترجمة.

مثال

ليكن لدينا الجملة التالية:

John play football

عند توليد الثلاثيات نحصل على:

Subject = John

Predicate = play

Object = football

وكان زمن الثلاثية ممتد من 00:02:22,643 --> 00:02:25,854

يتم تخزين ال Object على الشكل:

football#00:02:22,643 --> 00:02:25,854

6.17 تقييم ملفات RDF

- بعد أن تم تحميل ملفات نصوص الترجمة subtitle من خلال الوكيل البرمجي، تم تحويل نص الترجمة إلى ثلاثيات RDF Triples، ومن ثم توليد ملفات ال RDF المقابلة لهذه الثلاثيات التابعة لنص ترجمة معين، للتحقق من صحة ملفات ال RDF تم الاستعانة بمكتبة dotNetRDF والتي تؤمن عمليات التحقق المطلوبة.

6.18 رسم المخطط المفاهيمي Conceptual Graph

- بعد الحصول على الثلاثيات وتوليد ملقات ال RDF قمنا بخطوة إضافية توضيحية وهي رسم المخطط المفاهيمي Conceptual Graph المقابل للف RDF؛
- تم استخدام مكتبة GLEE لرسم المخطط المفاهيمي.

6.19 الأدوات البرمجية المستخدمة في استخراج الثلاثيات وتوليد RDF

1- في مرحلة التحسين على النص قبل استخرج الثلاثيات قمنا بالاستعانة بخدمة Coreference الموجودة ضمن مكتبة Stanford CoreNLP ، لكننا قمنا بتطبيق بعض التحسينات عليها لمعالجة بعض المشاكل مثل حصر هذه العملية فقط على الضمائر بكافة أنواعها مثل she, it, they، وإضافة 's في حال كان الضمير هو أحد ضمائر الملكية مثل his, her؛

2- ضمن مرحلة التحسين أيضا وخلال عملية lemmatization والتي تقوم بإرجاع الكلمة إلى أصلها بالاستعانة بمعجم مثل WordNet قمنا بالاستعانة بخدمة موجودة ضمن مكتبة Stanford CoreNLP؛

3- تم استخدام لغة البرمجة Java في تحقيق طريقة تحويل النص إلى ثلاثيات Offline؛

4- تم استخدام لغة البرمجة Java في تحقيق طريقة تحويل النص إلى ثلاثيات بالاستعانة ب Web Service تدعى Enrycher؛

5- تم استخدام لغة البرمجة C# في تحقيق عملية تحويل الثلاثيات إلى RDF بالاستعانة بمكتبة dotNetRDF؛

6- تم استخدام المحلل اللغوي JSRT للتحقق من صحة ملف SRT قبل استخراج الثلاثيات منه؛

7- تم استخدام dotNetRDF في تقييم ملفات RDF؛

8- تم الاستعانة بمكتبة GLEE لرسم ال Conceptual Graph؛

9- تم استخدام spotlight من أجل الربط مع dbpedia؛

10- الحصول على dbpedia types من خلال استعلام sparql والذي تتيحه لنا مكتبة dotNetRDF؛

7 تحقيق آلية للاستعلام عن المعلومات في المخزن الشخصي الذي تم إنشاؤه لصفوف البيانات

- المكتبة المستخدمة للاستعلام هي dotNetRDF وذلك عن طريق كتابة استعلامات بلغة sparql.

1-8) بحث عادي

- يتم إدخال الاستعلام المطلوب من قبل المستخدم من خلال واجهة بسيطة، يتم تحويل كلمات البحث إلى ثلاثيات RDF باستخدام الطرق المشروحة مسبقا؛
- بعد الحصول على ثلاثيات ال RDF، ومن أجل كل ثلاثية يتم تطبيق استعلام sparql والبحث ضمن مجلد ملفات ال RDF عن كل الثلاثيات المطابقة والشبيهة للثلاثية، نتيجة هذه المرحلة هي كل الثلاثيات الموجودة ضمن مجلد ملفات ال RDF والمشابهة لثلاثيات الاستعلام مع اسم الفيلم التابع لكل ثلاثية وزمن عرض المشهد الموجودة ضمنه الثلاثية؛
- من أجل كل ثلاثية من الثلاثيات المطابقة للاستعلام يتم الحصول على المشهد المقابل لها، في حال لم يكن موجود مسبقا ضمن مشاهد نتائج البحث يتم إضافته إلى المشهد، في حال كان موجود مسبقا يتم زيادة score هذا المشهد كأحد أساليب ال Ranking، حيث يتم ترتيب المشاهد حسب الثلاثيات الموجودة ضمن هذا المشهد، كل ما كان عدد الثلاثيات المحققة للاستعلام موجودة ضمن المشهد كان score هذا المشهد أعلى؛
- عرض نتائج الاستعلام والمتمثلة بالمشاهد "اسم الفيلم + زمن بداية المشهد + زمن نهاية المشهد" من خلال windows media player أو من خلال youtube.

(2-8 بحث متقدم

- **إمكانية وضع فلاتر**

يتم وضع فلاتر على أفلام البحث وبالتالي تقليص عدد الملفات التي سيتم البحث ضمنها بناء على الفلاتر المختارة:

- حيث من الممكن تحديد أفلام من نمط معين action, romance... وبالتالي البحث ضمن الأفلام التي تتبع هذا النمط فقط؛
- تحديد ممثلين محددين وبالتالي البحث ضمن الأفلام التي يوجد ضمنها هؤلاء الممثلين فقط؛

- تحديد لغة الفيلم وبالتالي البحث ضمن ملفات لغات الترجمة الخاصة بها هي من ضمن اللغات المحددة من قبل المستخدم؛
- تحديد سنوات محددة وبالتالي البحث ضمن الأفلام الموجودة ضمن السنوات المحددة من قبل المستخدم؛
- إعطاء المستخدم حرية اختيار domain للبحث عنه وبالتالي الحصول على الثلاثيات التي تنتمي إلى هذا ال domain حصرا.

- يتم إعطاء إشعار للمستخدم بكلمات البحث التي لم نحصل منها على triples وإعطائه حرية الاختيار فيما لو كان يريدها ك subject أو Predicate أو objects أو أي تركيبة منهم.
- إعطاء المستخدم حرية إدخال triple إلى الاستعلام مع نمطها "domain" خاص بها.
- إعطاء المستخدك حرية التعديل على ال triples الناتجة عن النص المستعلم عنه، من حذف وتعديل وما إلى ذلك.
- بعد إدخال keywords البحث والحصول على ال triples، من أجل كل ثلاثية ومن أجل كل كلمة ضمن الثلاثية، يتم أخذ أكثر synset مشهورة ضمن ال wordnet تحوي هذه الكلمة والحصول على مرادفاتها، ومن ثم الحصول على كامل التباديل المكنة وذلك بغية الحصول على نتائج مرادفة لكلمات البحث وليس مطابقة لها تماما.
- بعد الحصول على ثلاثيات ال RDF، ومن أجل كل ثلاثية يتم تطبيق استعلام sparql والبحث ضمن مجلد ملفات ال RDF عن كل الثلاثيات المطابقة والشبيهة للثلاثية، نتيجة هذه المرحلة هي كل الثلاثيات الموجودة ضمن مجلد ملفات ال RDF والمشابهة لثلاثيات الاستعلام مع اسم الفيلم التابع لكل ثلاثية وزمن عرض المشهد الموجودة ضمنه الثلاثية؛
- من أجل كل ثلاثية من الثلاثيات المطابقة للاستعلام يتم الحصول على المشهد المقابل لها، في حال لم يكن موجود مسبقا ضمن مشاهد نتائج البحث يتم إضافته إلى المشهد، في حال كان موجود مسبقا يتم زيادة score هذا المشهد

كأحد أساليب ال Ranking، حيث يتم ترتيب المشاهد حسب الثلاثيات الموجودة ضمن هذا المشهد، كل ما كان عدد الثلاثيات المحققة للاستعلام موجودة ضمن المشهد كان score هذا المشهد أعلى؛

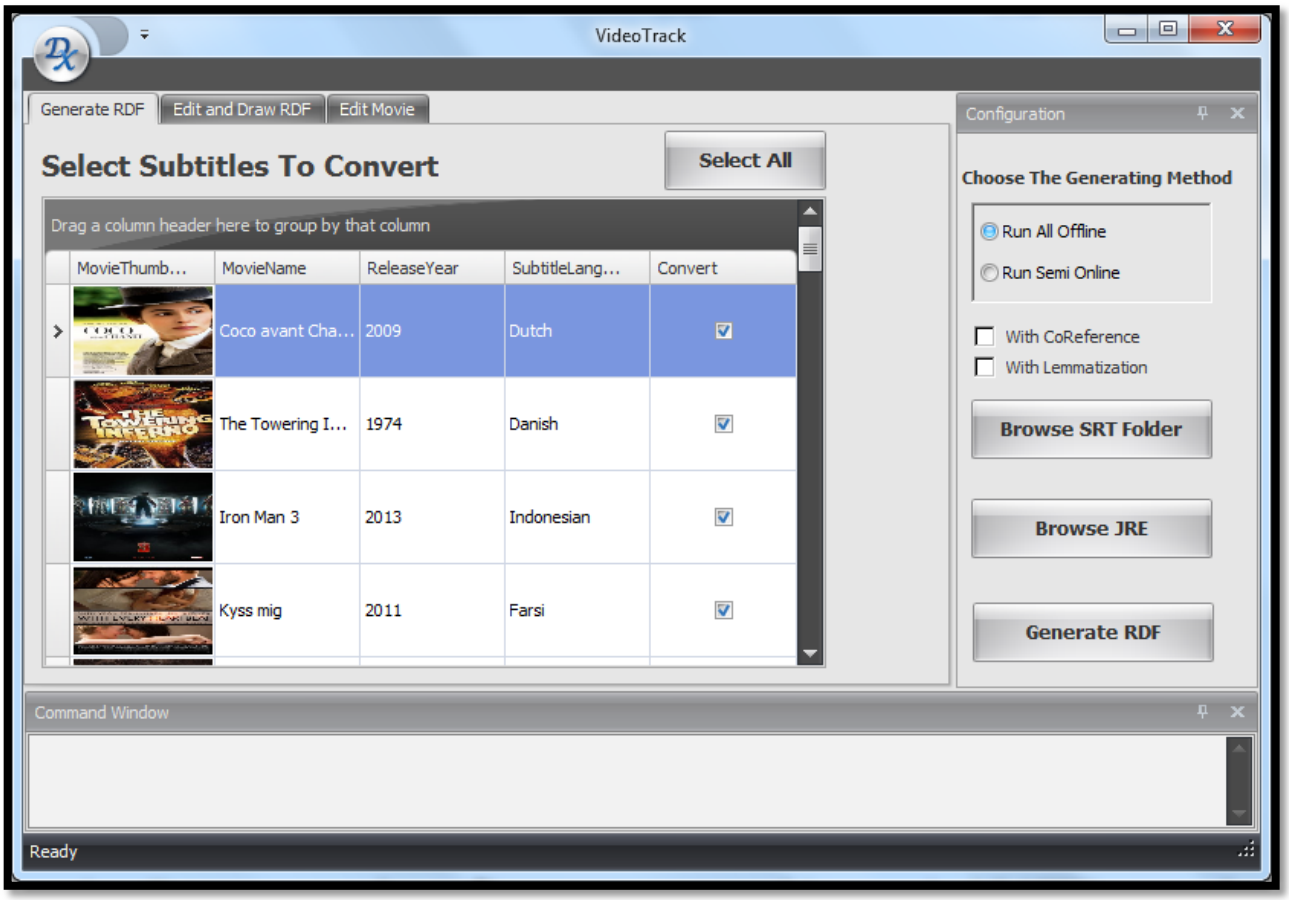
- عرض نتائج الاستعلام والمتمثلة بالمشاهد "اسم الفيلم + زمن بداية المشهد + زمن نهاية المشهد" من خلال windows media player أو من خلال youtube.

8 واجهات التطبيق الأساسية

8.1 Generate RDF

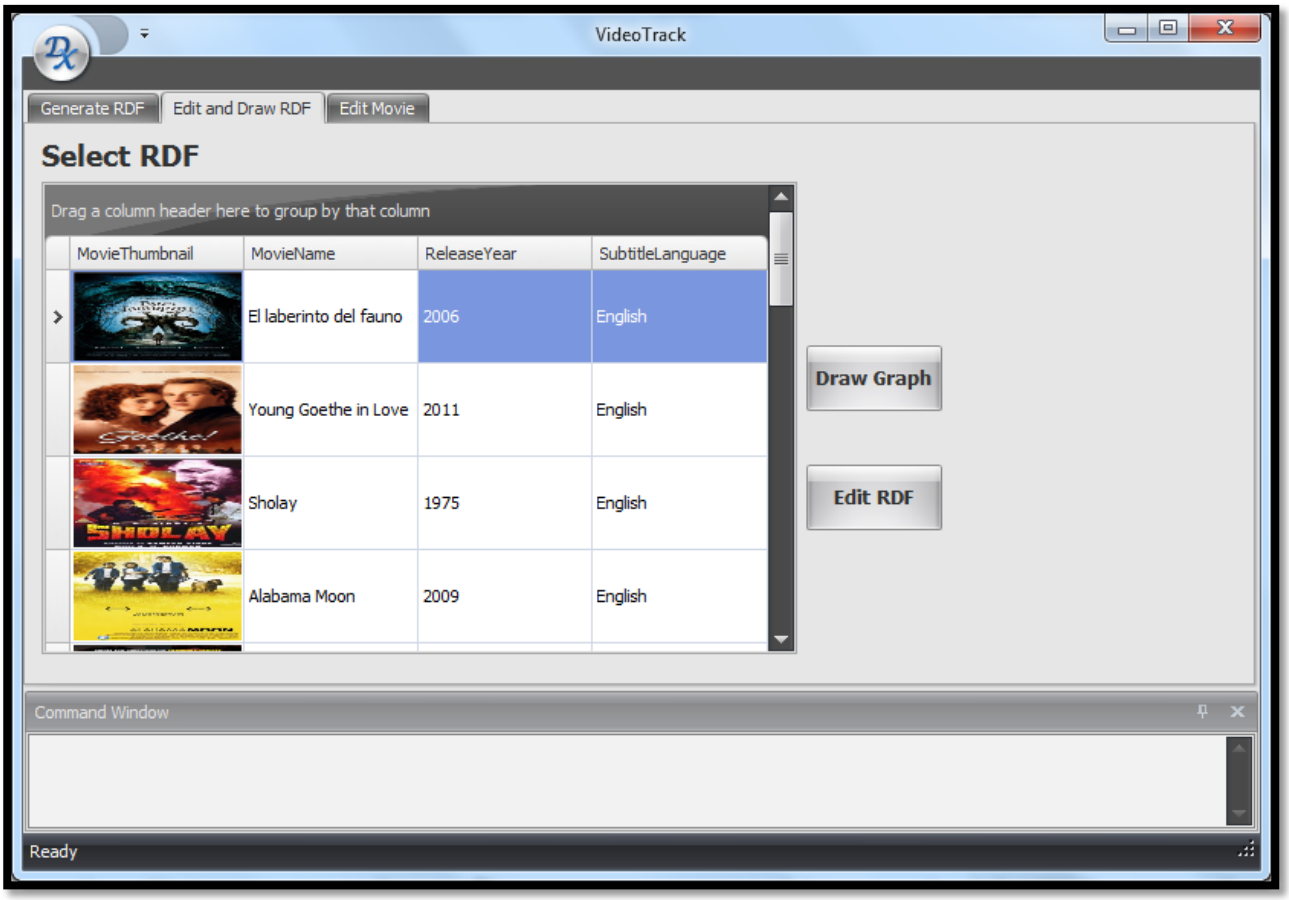
- يتم عرض قائمة بملفات الترجمة الموجودة والغير معالجة وذات اللغة الإنكليزية؛
- يقوم المستخدم بتحديد الملف– أو أكثر من ملف – المراد تحويله إلى RDF ؛
- يقوم المستخدم باختيار خوارزمية التحويل –جمع خوارزميات ال Offline– أو –جمع خوارزميات ال online– مع خوارزميات ال Offline؛

- في حال أراد المستخدم أن يتم معالجة النص المراد تحويله مع الأخذ بعين الاعتبار خاصية Coreference أو Lemmatization يقوم بتفعيلها؛



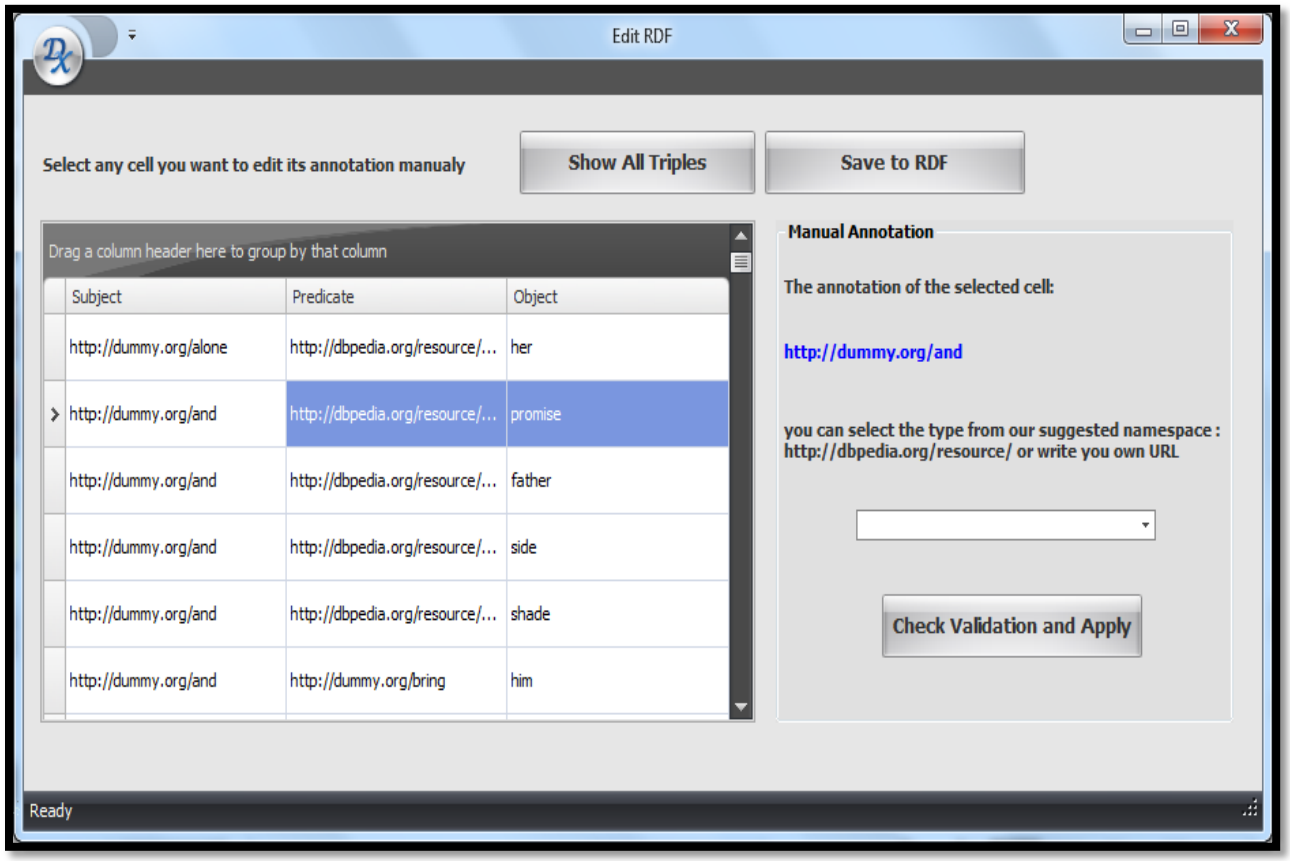
Edit And Draw RDF 8.2

- يتم عرض قائمة بملفات ال RDF المتاحة؛
- Edit RDF تظهر عندها واجهة تعديل ملف ال RDF؛
- Draw RDF تظهر عندها واجهة رسم ال conceptual graph الخاص بالملف المحدد.



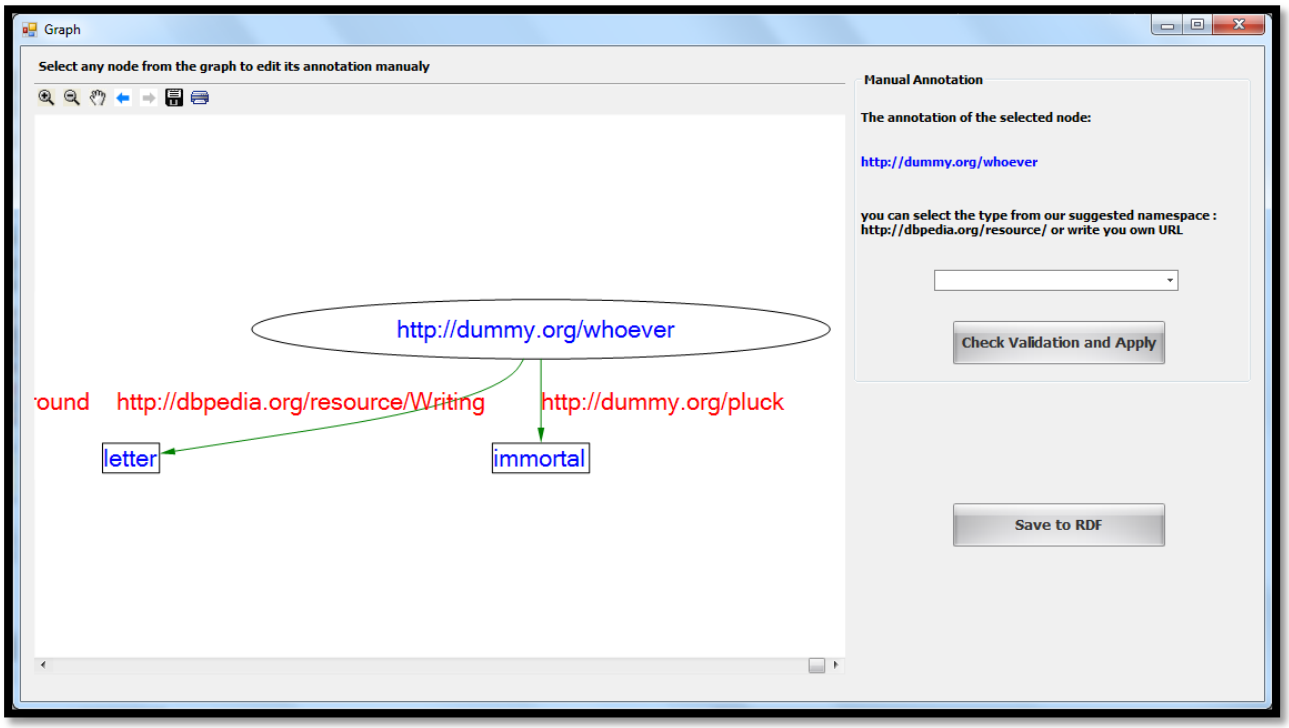
Edit RDF 8.3

- يتم إظهار الثلاثيات التي تحوي dummy Uri فقط أو كل الثلاثيات ضمن الملف حسب رغبة المستخدم؛
- يتم تحديد ال subject أو ال Predicate الذي يود المستخدم تعديله؛
- يظهر ال URI القديم للعنصر المختار باللون الأزرق على يمين الواجهة؛
- يدخل المستخدم ال URI الذي يريده أو يقوم باختيار Type من أنماط ال dbpedia؛
- يتم التحقق من Uri المدخل في حال لم يكن صحيح أو الصفحة غير موجودة يتم إعطاء تنبيه ورفض العنوان المدخل؛
- Save to RDF يتم حفظ التعديلات ضمن الملف المطلوب تعديله "إعادة بنائه مجددا".



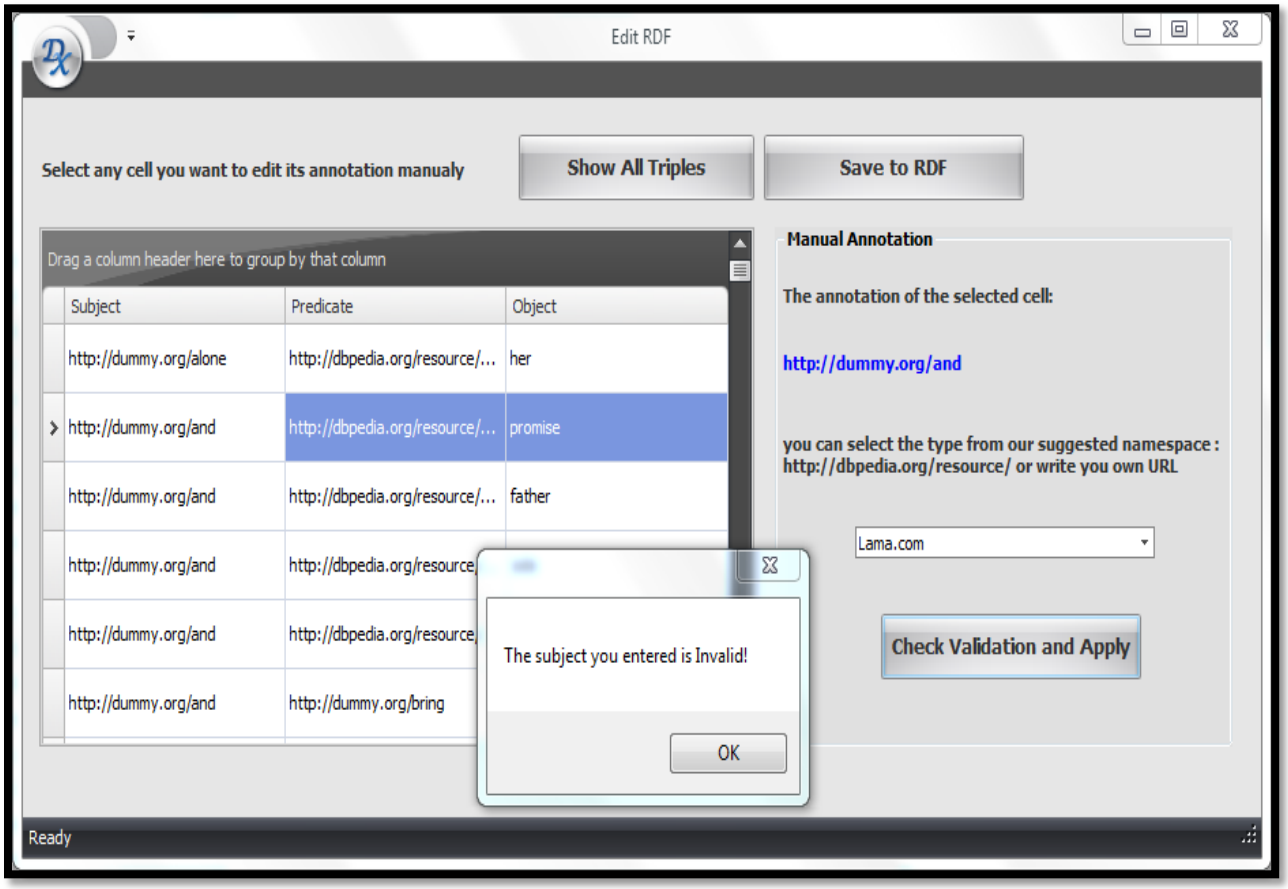
Draw Graph 8.4

- يتم رسم البيان الخاص بالملف المحدد؛
- يمكن تعديل ملف ال RDf من خلال البيان؛
- يتم تحديد العقدة أو الرابط الذي يريد تعديل ال Uri الخاص به؛
- بظهر الرابط القديم باللون الأزرق على يمين الواجهة؛
- يدخل المستخدم الرابط الجديد أو يختار أحد الأنماط المجردة؛
- يتم التحقق من صحة الرابط المدخل وإعطاء تنبيه في حال كان غير صحيح أو غير موجود؛
- Save to RDf حفظ التعديلات في حال كان المستخدم يريد تأكيد ذلك.



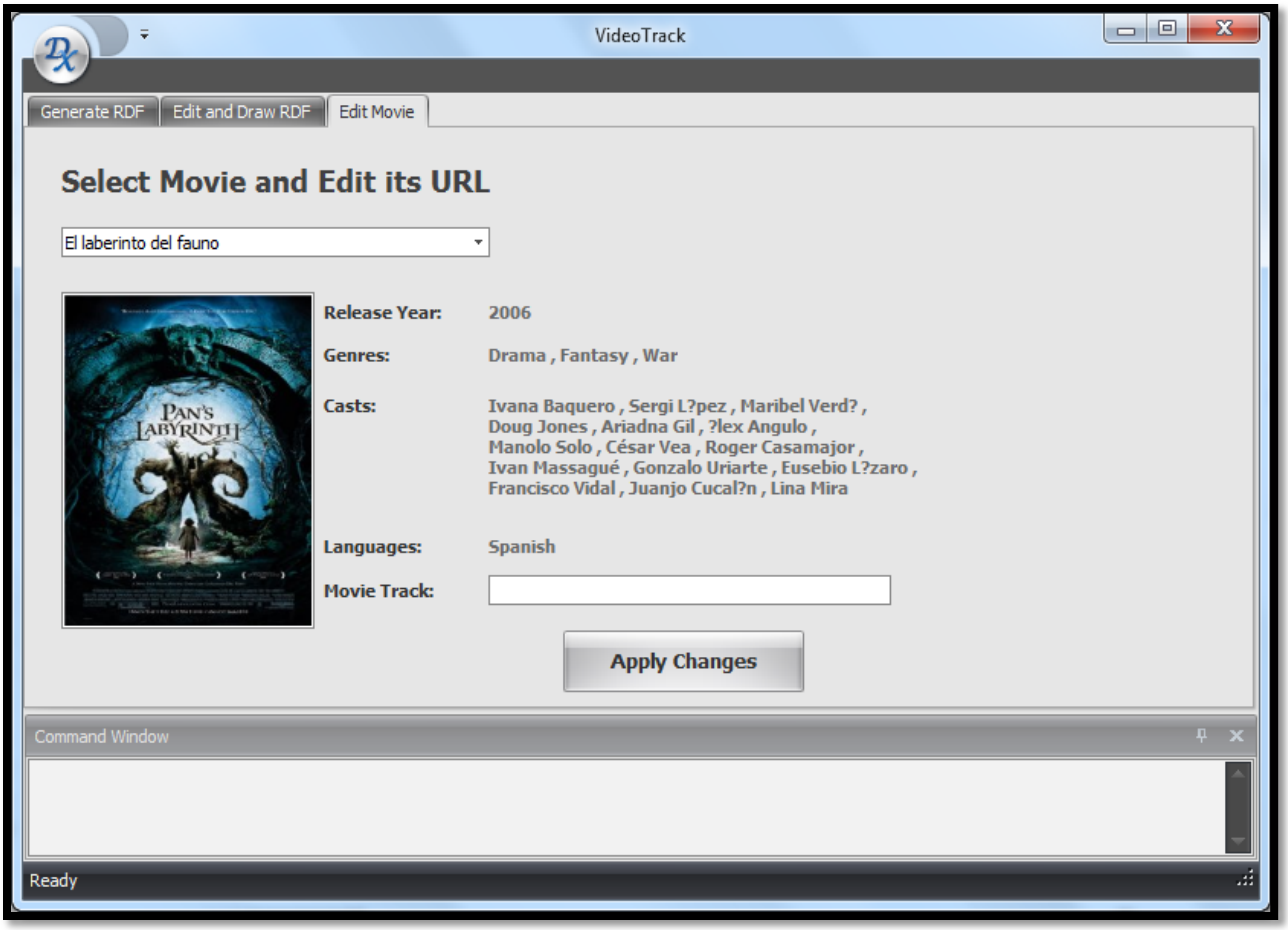
8.5 التحقق من العنوان المدخل

- يظهر الشكل عملية إدخل uri خاطئ وغير موجود في الأصل، نلاحظ كيفية ظهور تنبيه للمستخدم أن العنوان المدخل مرفوض.

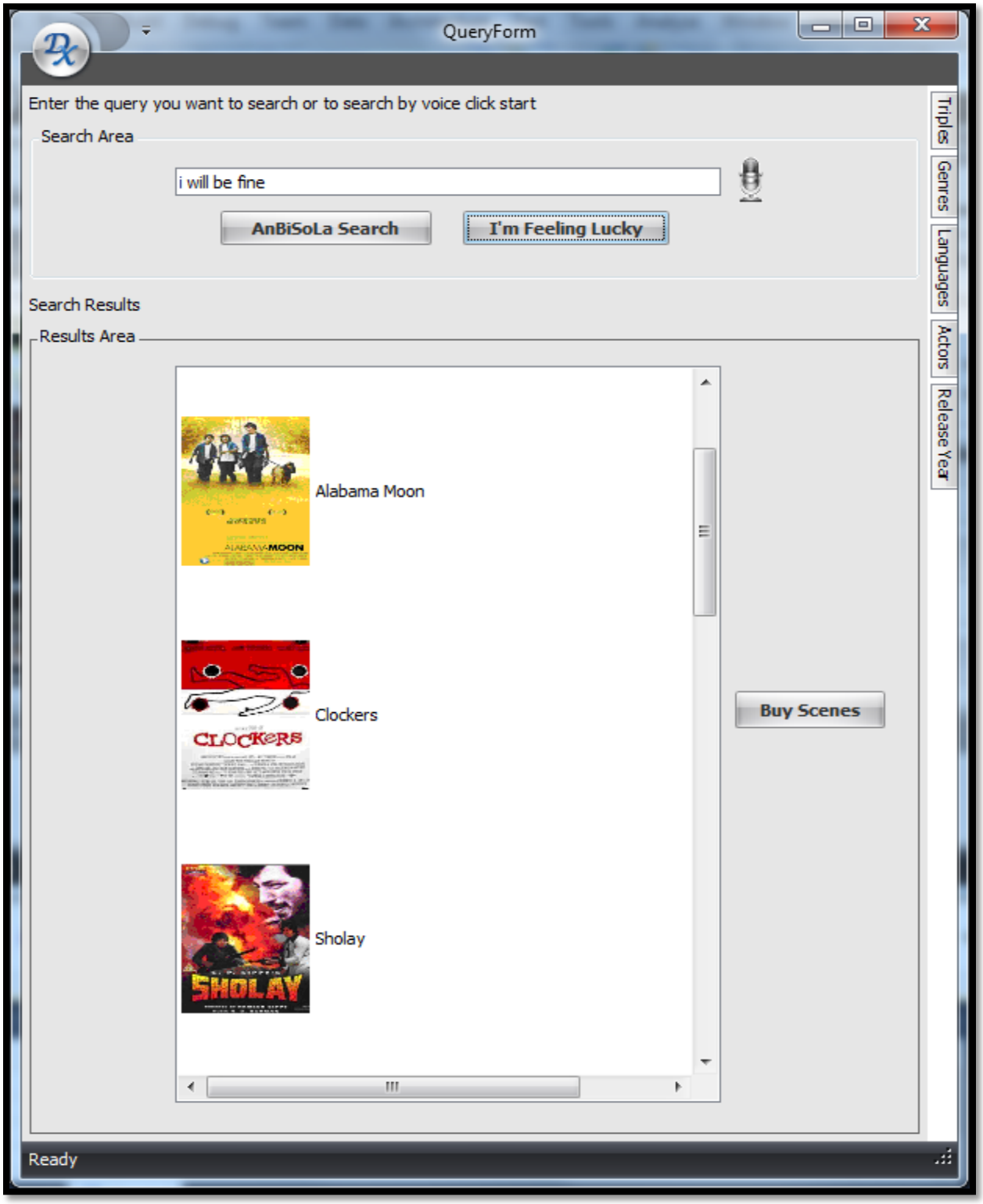


Edit Movie 8.6

- يتم عرض معلومات الفيلم المختار ليتم تحديد مكان تواجد الفيلم أو رابط الفيلم للاستفادة منه لاحقا في عمليات عرض النتائج.

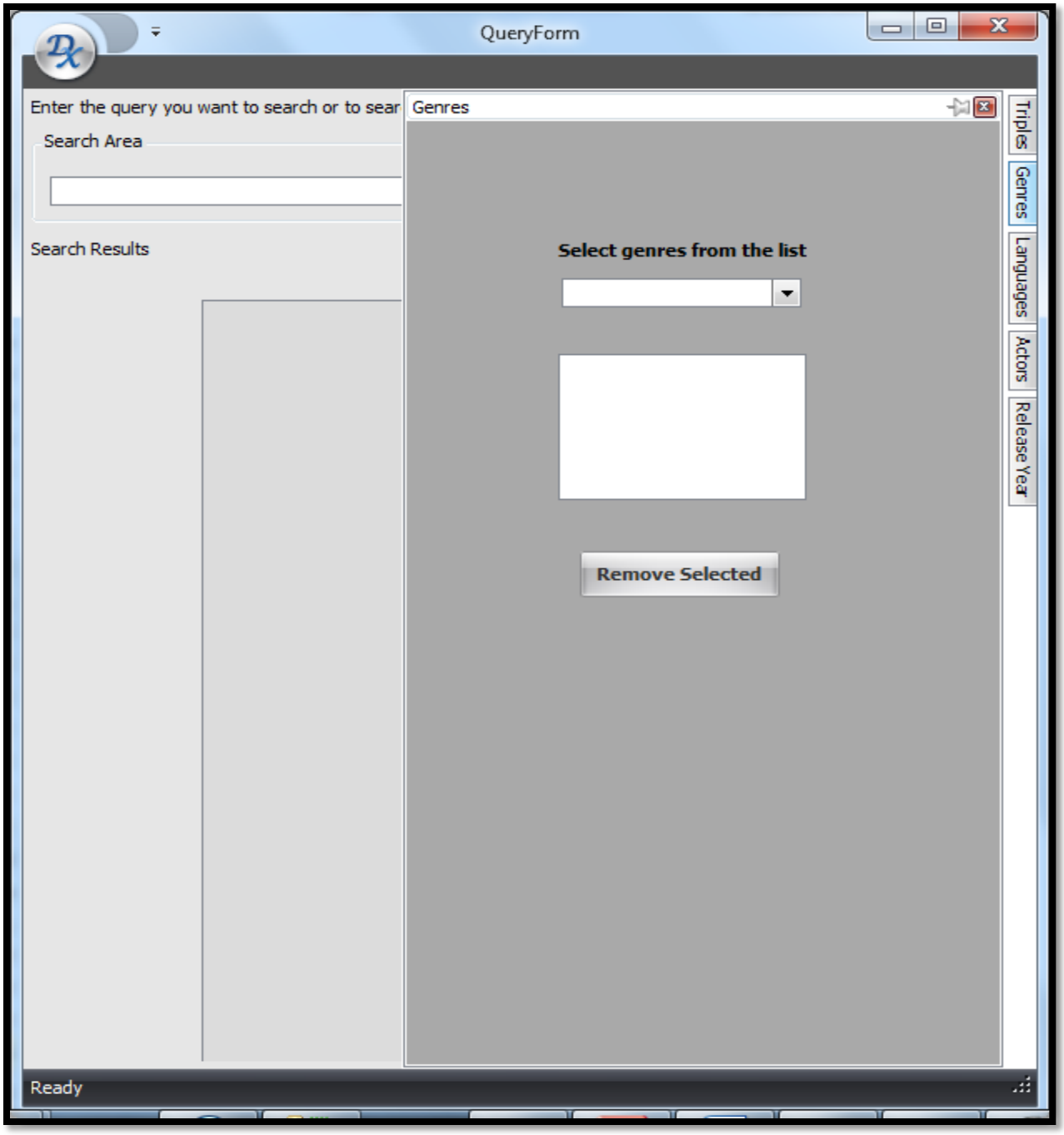


Query Interface 8.7



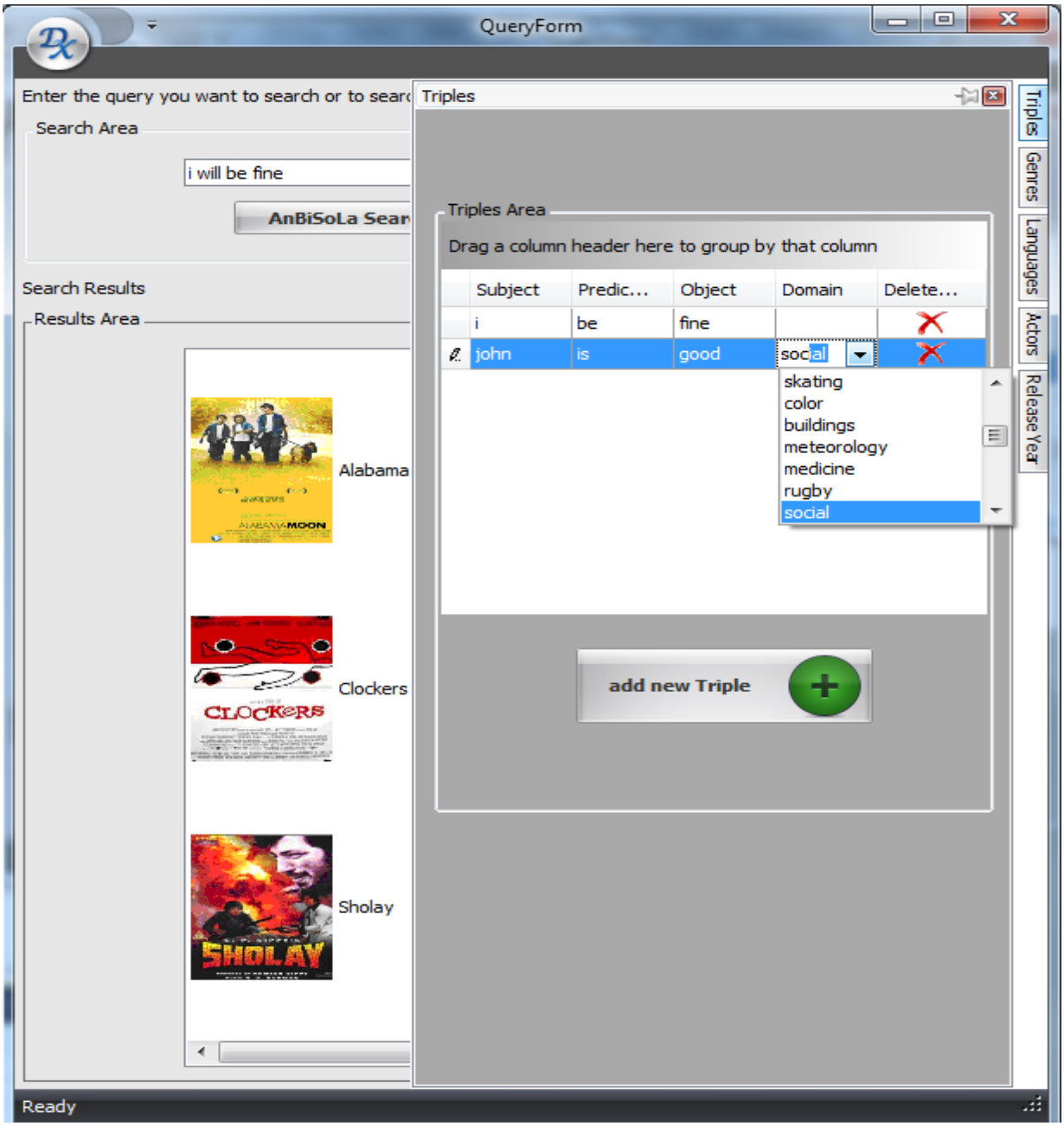
Choose Filters 8.8

- في البحث المتقدم يتم تحديد الفلاتر المطلوبة من نوع الفيلم والممثلين ولغة الفيلم وسنة إنتاج الفيلم.



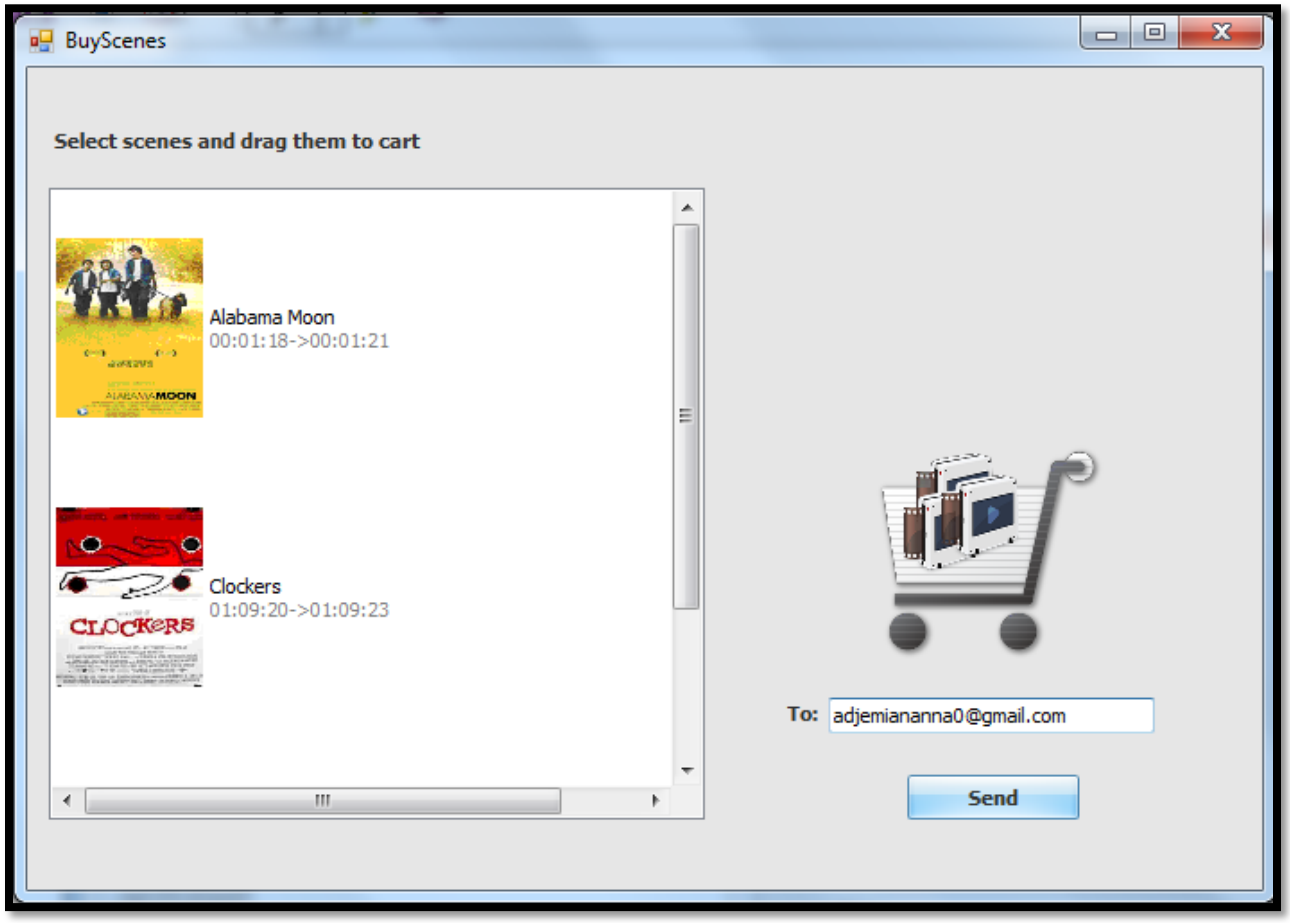
Edit and Add Triples 8.9

- يتم إظهار الثلاثيات الناتجة عن الاستعلام المدخل؛
- إعطاء المستخدم حرية التعديل على الثلاثيات الناتجة؛
- إعطاء المستخدم حرية إدخال ثلاثيات بحث جديدة مع ال domain الخاص بها.



Movie Display 8.10

Buy Scene 8.11



Tag Cloud 8.12

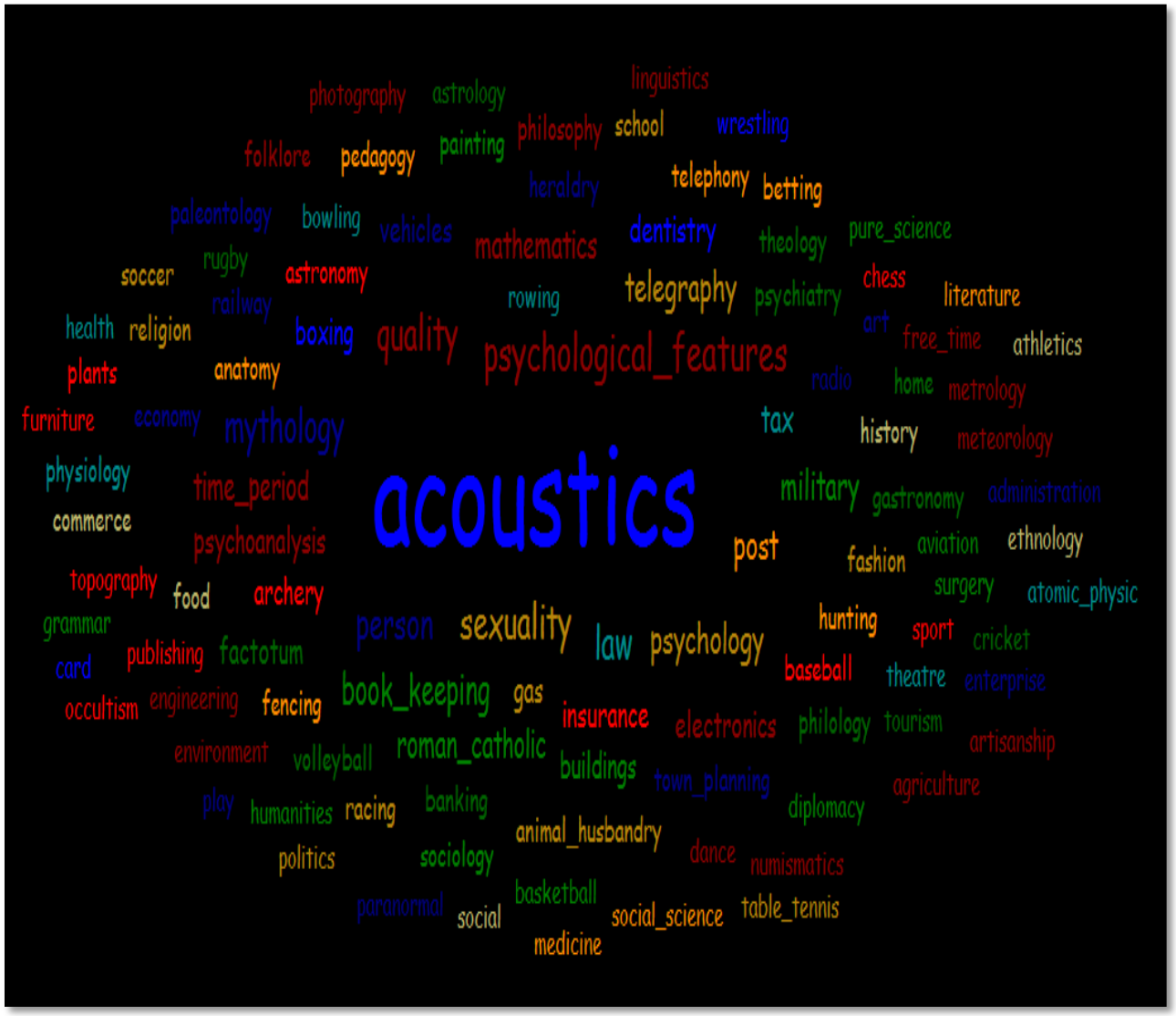
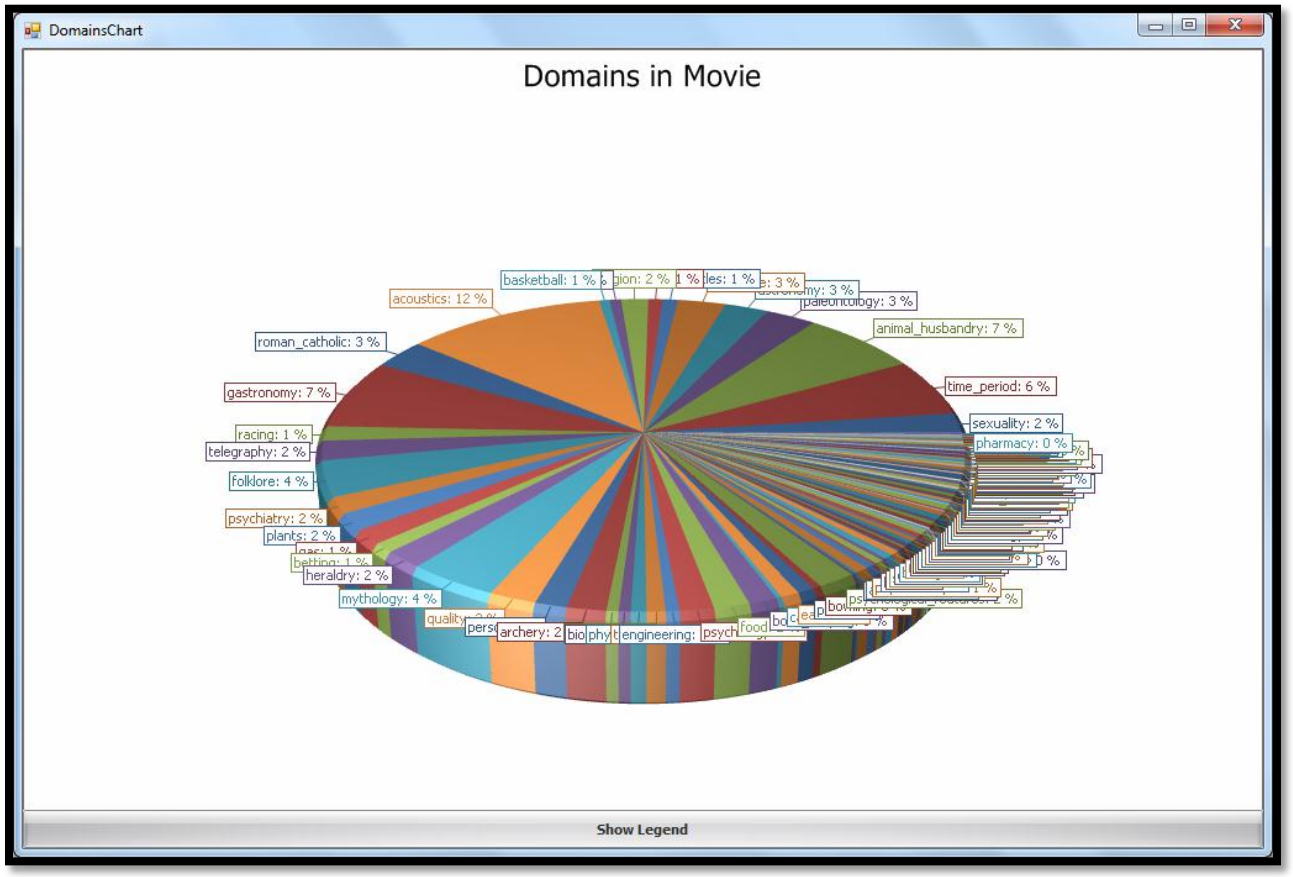


Chart 8.13


DomainsChart
Domains in Movie
basketball: 1 %
acoustics: 12 %
paleontology: 3 %
animal_husbandry: 7 %
roman_catholic: 3 %
time_period: 6 %
gastronomy: 7 %
sexuality: 2 %
pharmacy: 0 %
racing: 1 %
telegraphy: 2 %
folklore: 4 %
psychiatry: 2 %
plants: 2 %
heraldry: 2 %
mythology: 4 %
archery: 2
engineering:
Show Legend

9 المراجع المستخدمة

1. http://imdbapi.org/
2. http://htmlagilitypack.codeplex.com/
3. http://dotnetzip.herobo.com/DNZHelp/html/c1d9ee47-6e35-bcbe-3d5d-f9379d9e7dcc.htm
4. http://dotnetzip.herobo.com/DNZHelp/Index.html
5. http://api.themoviedb.org/
6. TRIPLET EXTRACTION FROM SENTENCES, Delia Rusu, Lorand Dali, Blaz Fortuna, Marko Grobelnik, Dunja Mladenia
7. Stanford typed dependencies manual – 2008 Marie-Catherine de Marneffe and Christopher D. Manning
8. http://stackoverflow.com/questions/9595983/tools-for-text-simplification-java/9606606#9606606
9. http://opennlp.apache.org/
10. http://nlp.stanford.edu/software/corenlp.shtml
11. http://nlp.stanford.edu/software/tagger.shtml
12. http://nlp.stanford.edu/software/dcoref.shtml
13. http://enrycher.ijs.si/