Un algoritme per detectar la relació temporal de dues paraules.

Treball de fi de Grau
Enginyeria Informàtica

Autor: Gerard Núñez Ruiz
Data: 21/06/2017
Director: Lluís Padró Cirera

Resum

La informació sobre la relació temporal entre dues paraules és una informació que nosaltres deduïm del text però és un camp actualment en investigació al nivell d'intel·ligència artificial. Aquest projecte està basat en el processament del llenguatge natural i té l'objectiu de implementar un sistema capaç de extreure les relacions temporals entre paraules. Aquest sistema utilitza una *support vector machine* amb un model prèviament entrenat.

Aquest sistema ha donat bons resultats comparat amb altres sistemes que donen solució al mateix problema i ha definit un sistema base amb el qual poder seguir la investigació sobre l'extracció de relacions temporals mitjançant el processament del llenguatge natural.

Agraïments

Durant el desenvolupament d'aquest treball de fi de grau han participat altres persones ja sigui de manera directa o indirecta, des de guiar-me i ajudar-me tècnicament com oferint el seu suport i ànims durant el seu desenvolupament.

En primer lloc, vull donar les gràcies al meu director de TFG, Lluís Padró, per oferir-me el seu suport des de el primer moment i guiar-me amb el seu coneixement davant de les dificultats amb les que m'he trobat. A més de ser comprensiu sempre que he tingut algun problema.

També vull donar les gràcies a la Facultat d'Informàtica de Barcelona per oferir-me una formació que m'han fet créixer tant a nivell personal com acadèmic.

I per últim agrair als meus familiars i amics per tot el suport i ajut rebut. Des de aguantar els meus discursos tècnics com opinar i intentar ajudar-me en aquells moments en que necessitava una empenta. Sobre tot als meus pares i germanes que m'han donat suport i s'han preocupat per l'estat del treball des de el minut 0.

A tots els que m'heu ajudat en aquest projecte, gràcies.

Índex

Gestió Econòmica

Implementació

Apèndix

Introducció

1 Abast del projecte

1.1 Formulació del problema

La interpretació del llenguatge natural és un problema d'intel·ligència artificial al qual s'estan enfrontant molts col·lectius d'investigació en els darrers anys. Aquesta interpretació s'utilitza per obrir un nou camí de comunicació entre l'usuari i l'aplicació, molt més còmode i normalitzat, ja que permet la comunicació amb un llenguatge molt flexible tal com és el llenguatge natural.

La interpretació del llenguatge natural te moltes aplicacions en diversos camps com la medicina, la industria, la economia i els videojocs es per això que no paren d'aparèixer noves aplicacions que utilitzen la interpretació del llenguatge natural com els assistents de Google, Microsoft o Apple.

El llenguatge natural el podem transmetre per text o per veu. En el cas de veu només s'ha d'afegir un pas previ de transformació del so en el text transcrit. En el nostre cas el canal d'entrada serà el mode text.

Del llenguatge natural ens interessa saber i analitzar moltes coses, però en aquest projecte ens centrarem en les relacions entre dues paraules. Concretament en la relació temporal de dos events. Aquests events seran verbs, noms o dates i el que volem saber és quin event succeeix abans, després, alhora o no tenen relació temporal.

La informació de relacions temporals es pot aplicar a diversos camps, per exemple, un sistema d'IA[1] que interpreta instruccions necessita saber quins events/accions ha de fer abans que d'altres per tal de dur a terme la tasca correctament. Per tant, un sistema extractor de relacions temporals pots ser molt útil en sistemes industrials automàtics, en els quals no has de transformar les instruccions del llenguatge natural a el format concret que entén la màquina, sinó que definint la tasca amb el llenguatge natural, es podria arribar a dur a terme per aquest sistema d'IA automàtic.

Com a conclusió, un sistema d'interpretació del llenguatge natural que detecti les relacions temporals de dos events podria aportar un nou sistema d'entrada i aprenentatge de tasques en maquines automàtiques molt més flexible i còmode.

1.2 Objectius del projecte

Ara que s'ha explicat el problema que volem solucionar amb aquest projecte, podem passar a analitzar els objectius que volem assolir.

L'objectiu principal d'aquest projecte es incloure un nou mòdul a la plataforma *Freeling* que afegeixi la informació de relacions temporals entre events.

Per tal d'assolir l'objectiu principal podem definir els següents:

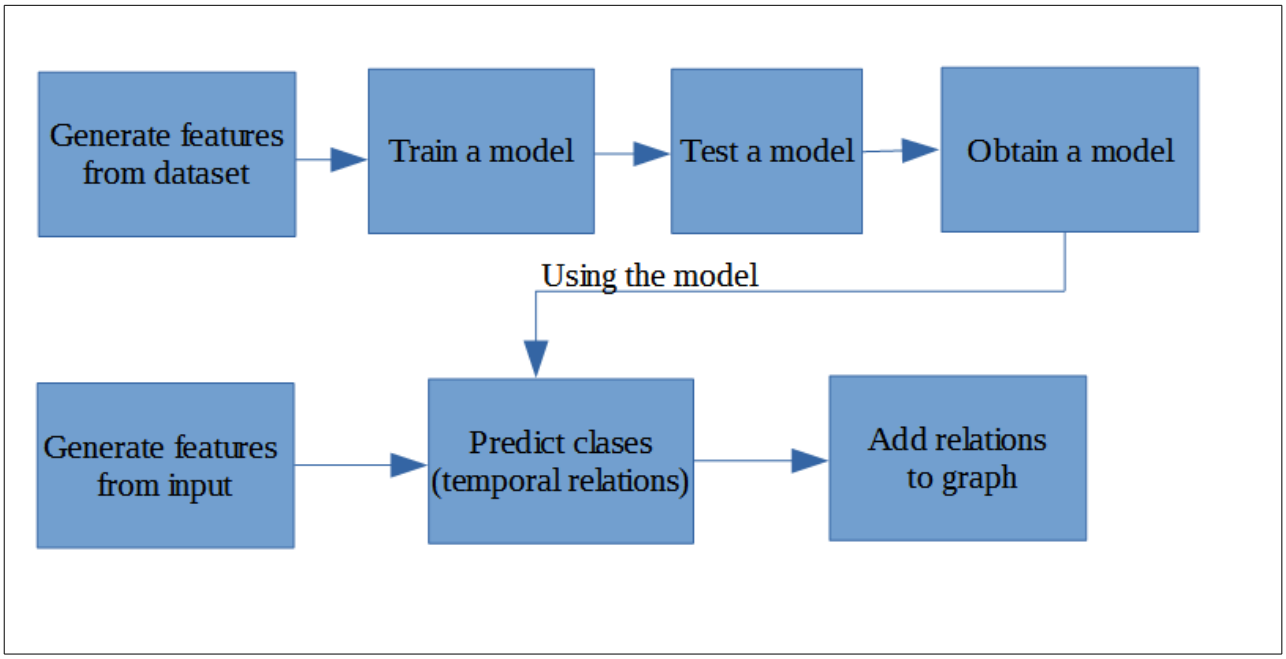
1. Estudiar i implementar una solució informàtica per la interpretació del llenguatge natural, concretament per l'extracció de relacions temporals entre events.

2. Fer una investigació sobre els millors models que ofereixen solució al problema.

3. Estudiar i participar en la plataforma *Freeling* per tal d'incloure el nou mòdul.

1 Intel·ligència artificial

Aquest és l'esquema de la solució:



1.3 Abast

Aquest projecte té dues partes ben diferenciades. La primera és la creació del millor model de classificació de relacions temporals entre events, i la segona, és la incorporació d'un nou mòdul al *Freeling* que utilitzi el classificador per afegir la informació temporal entre events al graf de relacions.

El desenvolupament d'aquest projecte parteix del *data set* de textos etiquetats que ofereix el concurs *TempEval-3* i s'utilitzarà com a conjunt de *train* per entrenar la *SVM*. Aquest entrenament definirà el model i el conjunt de *features* amb millors resultats en la classificació. Per tal d'obtenir aquest model es faran diferents proves canviant paràmetres de la SVM i conjunt de features. D'aquestes proves agafarem el model amb millors resultats de test per la segona part.

La segona part consistirà en incorporar un nou mòdul al *Freeling* que en l'anàlisi actual utilitzi el model obtingut en la primera part, per extreure les relacions temporals entre els events que trobi al text.

A partir del text d'entrada al *Freeling* es generaran els features associats al text d'entrada. Amb aquests features es farà una predicció per cada parella d'events i aquesta predicció, després de passar per un filtre de coherència, és a dir, que no es contradiguin dues o més relacions temporals, s'afegiran al graf de relacions temporals.

Com els *features* s'extreuen del *Freeling* la seva integració no ha de ser molt complexe.

Les dues parts s'hauran de desenvolupar sota un control de rendiment.

1.4 Possibles obstacles

Els possibles obstacles que poden aparèixer en aquest projecte són els següents:

L'obstacle principal és la complexitat del sistemes d'intel·ligència artificial. L'aprenentatge de models és complexe i potser hi han proves que no podem realitzar perquè falten recursos (memòria o temps d'execució). A més per que el model funcioni necessita un conjunt de *train*

prou gran, que podria provocar la falta de recursos. Per tal de evitar aquest problema l'aprenentatge s'executarà sobre els nodes del clúster del departament de CS[2].

En aquest projecte els errors de codi no seran molt significatius ja que no s'implementa cap algoritme complexe. Tot i així tant la generació de *features* com la integració a la plataforma *Freeling* estaran subjectes a proves exhaustives que assegurin el bon funcionament.

Com en qualsevol projecte de software el calendari pot arribar a ser un obstacle, ja que el temps està molt ajustat. Per tal d'evitar que el calendari sigui un obstacle es marcaran estrictament els objectius setmanals per garantir que es podrà arribar al objectiu final dintre del temps acordat.

1.5 Metodologia i rigor

La metodologia que aplicarem en aquest projecte és una metodologia de treball amb cicles curts, és a dir, amb fites petites que s'han d'assolir cada setmana. De la mateixa manera em reuniré amb el tutor del projecte setmanalment per veure l'estat del projecte i verificar les proves realitzades durant el desenvolupament.

En aquest cas el *feedback* del estat del projecte l'aportarà el tutor i serà ell qui acabi de concretar les fites de la setmana següent segons l'estat del projecte i el resultat de les proves de la setmana anterior.

Aquests cicles tindran una previsió des de un punt de vista general, però estaran subjectes a canvis particulars segons es vagui desenvolupant el projecte.

1.6 Eines per el desenvolupament

Per desenvolupar aquest projecte utilitzarem el *data set* de *TempEval-3* del que hem parlat en apartats anteriors. Aquest *data set* és un conjunt d'xmls amb els texts d'exemple etiquetats.

Per tal de tractar aquests xml utilitzarem un parser d'xml, el *pugixml*. Aquest parser és de codi obert i ens ofereix totes les funcionalitats que necessitem per parsejar l'xml d'entrada, tant el conjunt de train com el de test.

Tal i com hem dit abans el *features* els extraurem del *Freeling* per tant és una de les eines més importants d'aquest projecte. S'utilitzarà com analitzador de textos en les dues fases del projecte.

Com a classificador utilitzarem la *SVM libsvm* ja que és de codi obert i és el classificador inclòs al Freeling a més té bons resultats en solucions ja fetes d'aquest problema.

Per últim, com a llenguatge de programació s'utilitzarà *c++* per la generació de *features* i integració del mòdul ja que el Freeling està programat en *c++*. *Scripts* en *bash* i *python* per el tractament de fitxers. El control de versions del codi es farà mitjançant un repositori obert de *git*.

1.7 Seguiment del projecte

El seguiment del projecte es basarà en les reunions setmanals amb el tutor on s'analitzarà l'estat del projecte, les proves realitzades i les noves tasques a desenvolupar. Qualsevol desviació temporal en el desenvolupament de tasques s'ajustarà en aquestes reunions.

2 Computer Science

2 Context

Aquest projecte tracta de donar solució a un problema d'intel·ligència artificial referent a la interpretació del llenguatge natural. Per desenvolupar aquest projecte agafarem com a referencia la competició *TempEval-3* que consisteix en la etiquetació d'events i la extracció de relacions temporals.
Aquest projecte aportarà una nova solució aquest problema d'IA[3]. Més endavant profunditzarem en les explicacions més tècniques.

2.1 Actors implicats

Els actors implicats en el projecte, és a dir, aquelles persones o organitzacions que poden estar interessades en el projecte són:

Com a desenvolupador, dissenyador i beta tester seré jo l'encarregat ja que sóc l'única persona que porta el projecte.

El tutor d'aquest projecte és el Lluís Padró, el seu paper és el de supervisor del projecte. A més pot ajudar i guiar en el desenvolupament.

Un col·lectiu interessat en aquest projecte és tot aquell que està investigant sobre les relacions temporals entre dos events, ja que aquest projecte aportarà una nova solució i podrà ser testejada per tothom quan es publiqui a la plataforma *Freeling* de codi obert.

Per tant el col·lectiu que investiga sobre el llenguatge natural més interessat seran qui forma part de la plataforma *Freeling*.

A nivell d'usuaris no hi ha molta implicació més enllà del ús que pot rebre el *Freeling* per usuaris externs. Tot i així aquest projecte podrà ser utilitzat en altres aplicacions que si arribaran a una quantitat d'usuaris més alta.

3 Estat de l'art

3.1 Interpretació del llenguatge natural

La interpretació del llenguatge natural és una branca de la intel·ligència artificial que tracta computacionalment les llengües naturals. Les principals aplicacions o àrees de treball són la extracció d'informació, la cerca de respostes, la traducció automàtica i el reconeixement de la parla, entre d'altres. Aquestes aplicacions s'aconsegueixen a partir de l'anàlisi lèxica, l'anàlisi morfològica, l'anàlisi sintàctica i la interpretació semàntica. [11]

La plataforma *Freeling* que s'ha esmenat anteriorment realitza les tasques pròpies del PLN[4], encara que s'explicarà amb més profunditat al apartat següent. [1]

Per tant, del processament del llenguatge natural apareixen moltes tasques a realitzar per part de les aplicacions. En aquest projecte ens centrarem en la interpretació semàntica, concretament en la relació temporal de dos events.

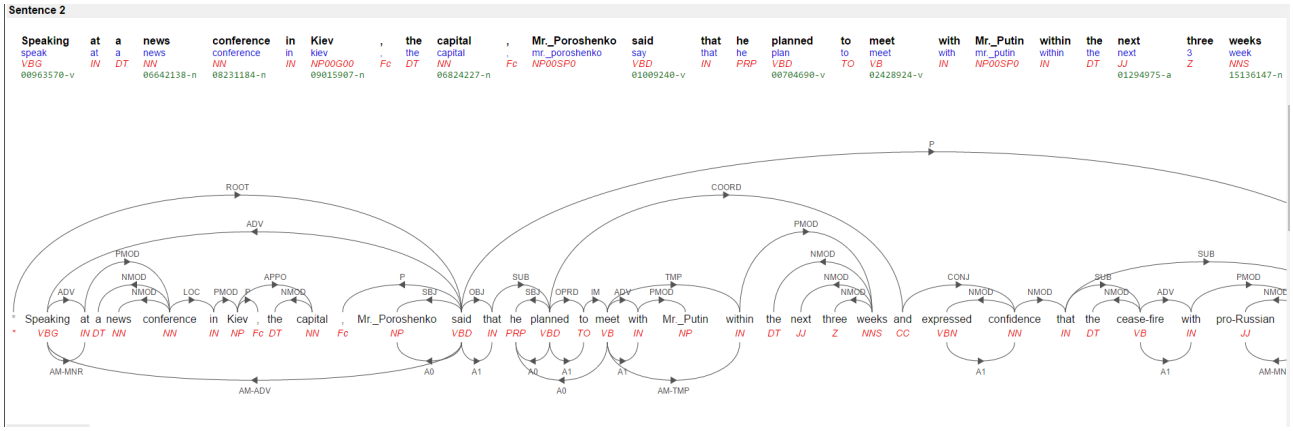
Aquest problema es presenta com un problema de classificació en el qual donat dos events hem de classificar-los en una classe temporal (BEFORE, AFTER, NONE, etc.). D'aquests classificadors en parlarem més endavant.

Sobre aquest problema podem trobar diferents solucions als papers que van sorgir del concurs *TempEval-3* que explicarem més endavant. [3][4][5][6][7]

3 Intel·ligència artificial.
4 Processament del Llenguatge Natural.

3.2 Freeling

Freeling és una plataforma on col·labora la Universitat Politècnica de Catalunya de codi obert basada en el processament del llenguatge natural. Aquesta plataforma ens ofereix un anàlisis profund d'un text a nivell lèxic, morfològic, sintàctic i semàntic. [1]



Graf extret de la demo online del Freeling

De tot aquest anàlisi ens ofereix un graf on es representen la informació lèxica i morfològica de cada paraula, les relacions sintàctiques entre sintagmes, les funcions dintre de les oracions i informació semàntica com la coreferència entre un nom i un pronom. I tot això t'ho ofereix en multidioma. [1]

La idea d'aquest projecte és incloure les relacions temporals entre paraules al graf esmentat.

3.3 TempEval-3

TempEval-3 és un concurs que té com objectiu avançar en el tractament de informació temporal. Aquest concurs té diferents parts/tasques a desenvolupar per els participants. La primera part és la detecció i etiquetació dels events i expressions temporals. Aquesta part no es desenvoluparà en aquest projecte. La segona part és la extracció de relacions temporals entre els events i expressions temporals. [2]

Aquest concurs ofereix els resultats que van aconseguir els participants en cada una de les tasques i alguns papers que van sorgir del concurs en els que donen més informació sobre la solució aplicada. [2]

Aquests papers seran referencia per el desenvolupament del projecte de tal manera que els *features* i classificador utilitzats en les solucions oferides pels participants seran estudiades i aplicades aquest projecte. [3][4][5][6][7]

3.4 Features

Un *feature* és una característica o peculiaritat d'alguna cosa. En aquest projecte definim *features* com les característiques que podem extreure d'una relació entre dos events, per exemple el PoS[5] dels events implicats, la distancia entre les dues paraules o el rol sintàctic que desenvolupen en la oració.

Els features utilitzats en aquest projecte seran extrets del complet anàlisis que el Freeling ofereix actualment. [1]

5 Part of Speech

Aquests *features* són utilitzats per el classificador per aprendre i crear un model. Un cop tenim el model, el classificador ha de predir nous vectors de *features* que representen la informació rellevant entre els dos nous events.

El conjunt de *features* pot ser molt complexa i variada però intentarem trobar un denominador comú entre els *features* proposats pels *papers* i els aportats per el tutor i per mi. Aquest conjunt és la base de la predicció, per tant es realitzarà un estudi sobre el conjunt de *features* que donin millors resultats, buscant el conjunt més petit amb el màxim percentatge d'encert. [3][4][5][6][7]

3.5 Classificadors

Hi han diversos tipus de classificadors, en el concurs *TempEval-3* s'utilitzen dos, classificadors basats en regles i support vector machines. En aquest projecte utilitzarem SVM[6]. [2]

Un classificador (SVM) és un sistema d'intel·ligència artificial que donat un conjunt de vectors codificats, vectors de *features* que formen diferents classes, crearà un model capaç de classificar/predir a quina classe pertany un altre vector amb la mateixa codificació. [10]

Aquest classificador, sobre aquest conjunt de classes, intenta crear un hiperplà que les separi de la millor manera possible mitjançant un vector de suport. El modelatge d'aquest model depèn dels paràmetres utilitzats. [10]

Els paràmetres més interessants que optimitzarem en aquest projecte són:

1. Kernel: funció que determina l'hiperplà, els més comuns són lineal, quadràtic, RBF[7].

2. C: "C" és el paràmetre de *marge* que indica la distancia entre els vectors de suport. Aquesta C es tradueix en la permissivitat d'error en la classificació. [10]

Una SVM és un sistema de classificació flexible que ens permet tenir diferents conjunts d'entrenament codificats amb diferents conjunts de *features* i testejar els models per veure quin dóna millors resultats. [10]

La elecció d'una SVM com classificador per aquest projecte s'ha basat en els bons resultats que ofereix i la fàcil integració d'aquest sistema en el *Freeling* ja que la plataforma té una SVM incorporada. [1][9][10]

3.6 Conclusions

Com a conclusió podem dir que aquest sistema d'intel·ligència artificial ha de resoldre un problema d'optimització d'un model.

En el nostre cas utilitzarem una SVM com classificador per les raons abans citades i optimitzarem el model emprat per la classificació segons el conjunt de *features* i els paràmetres del classificador.

6 Support Vector Machine
7 Radial Base Functions

Planificació temporal

4 Descripció de les tasques

El projecte té una duració aproximada de 8 mesos, del 1 de octubre del 2016 al 31 de maig del 2017.

En aquest apartat explicarem les tasques a desenvolupar en aquest projecte. El projecte té dues partes ben diferenciades on una depèn de l'altra. Aquestes tasques, amb les subtasques corresponents, s'explicaran a continuació.

4.1 Investigació i anàlisi inicial

La primera part del projecte consistirà en una petita investigació dels treballs anteriors relacionats amb aquest projecte i la creació del disseny i planificació inicial.

4.2 Creació del model

La creació del model és la primera tasca principal d'aquest projecte, sense aquesta part no es podrà començar la segona.

La creació del model consisteix en l'entrenament d'una SVM[8], a partir del conjunt de training, per modelar un classificador de parelles d'events segons la seva relació temporal.

Aquesta tasca ocuparà la major part del projecte i consistirà en les següents subtasques:

4.2.1 Creació de features

Els vectors de features són la base d'una SVM amb els quals aprèn i prediu. El conjunt de features pot ser molt divers i de la bona elecció depenen els resultats obtinguts.

Es per això que es dedicarà una bona part del temps en l'anàlisi de conjunt de features per tal de formar un conjunt suficientment bo. Per desenvolupar aquesta part, s'analitzaran conjunt de features utilitzats per altres estudis i es plantejaran nous amb el tutor del projecte.

Un cop marcat el conjunt de features inicial, s'ha d'implementar un programa en c++ que donat un conjunt de textos etiquetats (xml) generi els features associats a cada relació entre events.

Com la qualitat del conjunt de features depèn de la qualitat del model generat a partir d'aquest conjunt de features, el generador de features ha de ser suficientment flexible per poder afegir i treure features en el moment de generar models.

Per tant, la revisió d'aquesta tasca es farà generant un conjunt de models i provant-los amb un conjunt de development. Aquest procès s'explicarà en l'apartat següent.

4.2.2 Entrenament de models

Un cop definit un o més conjunts de features, els utilitzarem per entrenar models de predicció. L'objectiu d'aquesta tasca és aconseguir entrenar el model optim. La qualitat del model es mesura en qualitat de predicció, és a dir, quantitat de exemples classificats correctament.
Aquesta qualitat depèn del conjunt de features i els paràmetres utilitzats en l'aprenentatge de la SVM. Per tant, entrenarem la SVM amb diversos conjunts de features i diversos valors de paràmetres.

8 Support Vector Machine

El conjunt de features optim és aquell que amb un nombre mínim de features per relació aconsegueix la màxima precisió en la predicció i els paràmetres que s'optimitzaran en aquest projecte es van explicar en apartats anteriors.

Per entrenar els models utilitzarem el clúster del departament de CS[9] ja que la gran quantitat d'exemples i la complexitat en la generació del model i predicció no ens permet executar-la en màquines ordinàries. Això provocarà que aquesta subtasca sigui la que major temps requerirà ja que algunes proves potser triguen 1 setmana en executar-se.

Un cop generats els models es faran predir contra un conjunt de development. Sobre aquestes prediccions es calcularà la precisió i el recall per quantificar la qualitat d'aquest model.

Un cop optimitzat contra el conjunt de development, es mesurarà la qualitat del model contra un conjunt de test diferent del de development per tal de definir la qualitat real. Sense aquest test el model optim anterior seria aquell que contestes exactament el que el development espera però funcionaria malament contra els exemples reals als que s'enfrontarà el model.

Amb aquest doble test ens assegurem de que el model està correctament testejat.

Un cap tenim un model optimitzat podem passar a la següent tasca principal.

4.3 Creació del nou mòdul al Freeling

La creació del nou mòdul al Freeling consistirà, bàsicament, en incorporar el model optimitzat en la tasca anterior i afegir la informació que aquest model aportarà en l'anàlisi que ja incorpora el Freeling.

Per tal de realitzar aquesta tasca, s'haurà d'utilitzar el mòdul de classificador que ja incorpora el Freeling i generar informació temporal per cada parella d'events.

El procés de predicció consistirà en:

1. Generar els features de cada parella d'events del text d'entrada.
2. Predir a quina classe pertany cada relació, és a dir, predir la relació temporal de cada parella d'events.

Un cop obtinguda la predicció, hem d'incorporar-la al graf generat pel Freeling. Aquesta incorporació potser es fa integrada al graf existent o es generarà un altre per tal de no carregar amb massa informació l'existent.

En aquest graf de relacions temporals apareixeran les parelles d'event connectats entre si indicant la relació (BEFORE, AFTER, etc.).

Aquesta tasca depèn totalment de l'anterior ja que sense model no es podrà realitzar, però un cop generat el primer model, encara que no sigui l'optim, ja es podrà començar a fer proves i desenvolupar aquest mòdul encara que no sigui amb el model definitiu.

9 Computer Science

4.4 Test Final

Un cop integrat el nou mòdul al Freeling caldrà fer uns tests finals per verificar i quantificar la qualitat de la solució final. A partir d'aquest apartat s'extrauran coses a millorar per futurs projectes.

4.5 Documentació final

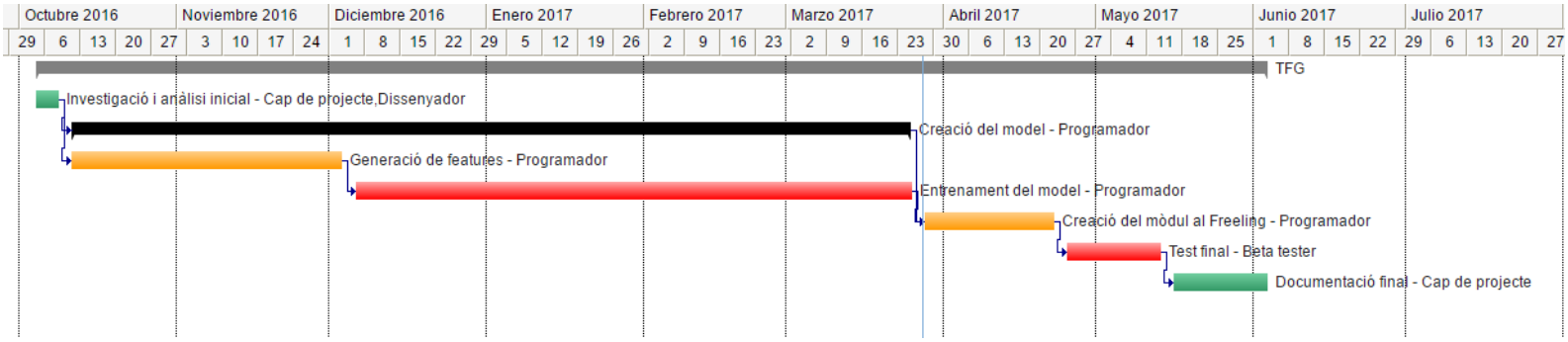
La part final del projecte es dedicarà a la documentació de la memòria, documents tècnics i codi relacionat amb aquest projecte.

4.6 Duració aproximada

La duració aproximada d'aquest projecte s'ha definit de la següent manera:

Tasca	**Duració (setmanes)**
Investigació i anàlisi inicial	1 setmanes
Generació de features	8 setmanes
Entrenament del model	16 setmanes
Creació del mòdul al Freeling	4 setmanes
Test final	3 setmanes
Documentació final	3 setmanes

4.7 Diagrama de Grantt



		Nombre	Duración	Inicio	Fin	Predecesoras	Recursos
0		**TFG**	175d	**03/10/2016**	**02/06/2017**		
1		Investigació i anàlisi inicial	1s	03/10/2016	07/10/2016		Cap de projecte,Dissenyador
2		**Creació del model**	24s	**10/10/2016**	**24/03/2017**	**1**	**Programador**
3		Generació de features	8s	10/10/2016	02/12/2016	1	Programador
4		Entrenament del model	16s	05/12/2016	24/03/2017	3	Programador
5		Creació del mòdul al Freeling	4s	27/03/2017	21/04/2017	2,4	Programador
6		Test final	3s	24/04/2017	12/05/2017	5	Beta tester
7		Documentació final	3s	15/05/2017	02/06/2017	6	Cap de projecte

Les tasques verdes són les de risc baix, les taronges de risc mitjà i les vermelles de risc alt.

Com podem veure a la taula de tasques la primera i ultima tasque formen part de la fase «Poject Manag. & People». Tot i que també tindrà rellevància en les reunions de seguiment durant les tasques de desenvolupament.

Per altre banda el beta tester intervé en la tasca de test final però també ha de testejar el model durant el seu entrenament.

4.8 Recursos

Els recursos necessaris per aquest projecte seran diferents eines de hardware i software i recursos humans.

4.8.1 Recursos Humans

Com a recursos humans necessitarem desenvolupar 4 rols, encara que una mateixa persona pugui fer més d'un rol.

- Cap de projecte
- Dissenyador
- Enginyer del software
- Beta tester

4.8.2 Hardware

- Portàtil Asus: Pel desenvolupament de codi.
- Clúster departament CS: Per l'entrenament de la SVM.

4.8.3 Software

- Editors de text (Sublime-text, VIM): Per el desenvolupament de codi.
- Sistema operatiu Linux: Per connectar-nos al clúster i desenvolupar el codi.
- Libre Office 2016: Per elaborar la documentació.
- Trello: Per la gestió de tasques.
- Git: Pel control de versions del codi.
- Freeling: Per l'anàlisi morfosintàctic i semàntic.
- Libsvm: Per la creació dels models.
- Pugixml: Pel parseig d'xmls.
- Llibreries estandard de c++, python i bash: Per el desenvolupament de codi.

5 Valoració d'alternatives i pla d'acció

Les desviacions que poden aparèixer en aquest projecte són, bàsicament, retards a l'hora de solucionar les tasques i per tant endarrerir les tasques que depenen d'aquesta. Per tal de solucionar aquest problema s'ha plantejat el temps estimat amb un marge que accepta cert grau de retard. A més, tant les proves d'optimització del model com les proves de test s'adaptaran al temps de resposta de cada una, és a dir, si una prova de model triga una setmana en realitzar-se es faran menys proves que si triguen 2 dies.

La tasca més crítica és la creació del model ja que sense un model amb un error acceptable no funcionarà correctament el nou modul del *Freeling*, podríem dir que és el pilar del projecte, es per això que la majoria del temps es dedicarà en aquesta recerca i les altres tasques s'acotaran segons les desviacions que sorgeixin en les tasques critiques. Per tal de no ajustar dues tasques critiques s'han distribuït de manera que no hi han dues critiques consecutives al *Grantt*.

No obstant, encara que una tasca no estigui 100% solucionada, com per exemple el model encara no és optim, es poden començar a desenvolupar tasques que depenen d'aquesta amb els resultats intermedis.

El control de les desviacions es realitzarà en les reunions setmanals que es faran amb el tutor del projecte. Aquestes reunions aniran marcant la dedicació a cada tasca segons les desviacions que apareguin durant el desenvolupament.

Donat que un cop generem el primer model, es podran desenvolupar les següents tasques, aquest projecte podrà finalitzar en el temps establert. El que no es pot assegurar es si s'aconseguirà una solució optima però s'intentarà, amb el pla d'acció anterior, aconseguir la millor solució possible.

Gestió econòmica

6 Identificació dels costos

Totes les tasques explicades en els apartats anteriors tenen un cost de recursos humans, hardware, software i indirectes associat als recursos utilitzats en la planificació. D'aquests costos en derivarem un pressupost.

El pressupost calculat deriva de les tasques presentades en el diagrama de Grantt.

7 Estimació dels costos

7.1 Costos directes

7.1.1 Recursos humans

Aquest projecte té costos en recursos humans tot i que només el desenvolupa una persona i el seu tutor.

Rol	**Hores**	**Preu per hores**	**Cost total**
Cap de projecte	110	40€/h	4.400€
Dissenyador	70	35€/h	2.450€
Enginyer de software	360	35€/h	12.600€
Beta tester	21	25€/h	525€
Total	**561**		**19.975€**

Aquests costos s'han calculat assumit el preu per hora indicat a la taula.

7.1.2 Hardware

A més, aquest projecte tindrà uns costos associats al hardware utilitzat (Amortització). A continuació es mostra una taula que detalla aquests costos.

Producte	**Preu**	**Vida útil**	**Amortització**
Asus K550CA	469€[10]	4 años	5,35€[11]
Clúster CS[12]	**[13]	**	**
Total			**5,35€**

10 Preu extret de www.pccomponentes.com
11 469€/35040 hores (4 anys) * 400 hores (d'ús) = 5,35€
12 Computer Science
13 No he pogut calcular el cost associat d'aquest hardware ja que jo he tingut accés gratuït i no he trobat informació en relació el preu i vida útil.

7.1.3 Software

Les eines de sotfware utilitzades en aquest projecte són totes gratuïtes, tot i això afegeixo un llistat.

- Editors de text (Sublime-text, VIM).
- Sistema operatiu Linux.
- Libre Office 2016.
- Trello.
- Git.
- Freeling.
- Libsvm.
- Pugixml.
- Llibreries estandard de c++, python i bash.

7.2 Costos Indirectes

Els costos indirectes d'aquest projecte consistiran en el transport per les reunions de seguiment i la quota mensual d'internet. La despesa d'electricitat no es tindrà en compte ja que el desenvolupament d'aquest projecte es realitzarà a un ordinador que ja s'utilitza diàriament, per tant no afegirà una despesa extra.

Despesa	**Preu per mes**	**Duració**	**Cost total**
Transport	47,33€ [13]	9 mesos	425,97€
Internet	30€ [12]	9 mesos	270€
Total			**695,97€**

7.3 Pressupost total

Despesa	**Preu**
Costos Humans	19.975€
Costos Hardware	5,35€
Costos Software	0€
Costos Indirectes	695,97€
Total	**20.676,32€**

8 Control de gestió

Les desviacions que poden afectar al pressupost són aquelles que augmentin la duració de les tasques i per tant augmentin els costos en recursos humans i costos indirectes.

Per tractar aquestes desviacions es faran reunions de seguiment que permetran ajustar els temps establers, i per tant el pressupost, de cada una de les tasques del Grantt. És a dir, es portarà un control de les tasques de major risc per establir el màxim temps de dedicació en tasques de menys risc.

Amb aquestes mesures es garanteix que la variació de pressupost no podrà arribar a ser molt gran i que es finalitzarà el projecte en el temps establert.

Per quantificar aquestes desviacions es faran servir indicadors de desviació per cada cost detallat anteriorment. A continuació es mostra una taula resum amb aquests indicadors:

Indicador	Fórmula
Desviació de mà d'obra en preu	(cost std – cost real) * consum hores real
Desviació de mà d'obra en consum	(consum hores std – consum hores real) * cost std
Desviació total en imports	Total costos std – total costos reals
Desviació total de costos fixos	Total costos pressupostats – total costos reals

9 Sostenibilitat

A continuació es mostra la matriu puntuada sobre sostenibilitat i compromís social en el fita inicial i posteriorment es descriuen detalladament.

	PPP	Vida Útil
Ambiental	Consum del disseny	Petjada ecològica
	3/10	17/20
Econòmic	Factura	Pla de viabilitat
	8/10	15/20
Social	Impacte personal	Impacte social
	8/10	15/20
Rang sostenibilitat	19/30	47/60
	66/90	

9.1 Dimensió Ambiental

Respecte el consum del disseny, s'utilitzarà un ordinador portàtil tant per el desenvolupament del projecte com per redactar la documentació. Els dispositius portàtils tenen molts components enfocats al baix consum per tal d'augmentar la duració de la bateria, es per això que el consum dels portàtils és menor que els ordinadors de sobretaula.

Encara que el clúster de CS, és un ordinador amb molts processadors que està encès 24 hores al dia. Això implica un gran consum. Tot i això aquest projecte necessita la potencia que pot oferir el clúster per calcular certes tasques.

Respecte la petjada ecològica, al ser un producte de software la seva petjada ecològica no serà molt gran. Com a molt podem dir que tindrà impacte si aquest software s'inclou en nous dispositius, que per tant tindran el seu cost ecològic de fabricació i un cop acabi la seva vida útil.

9.2 Dimensió Econòmica

Respecte la factura, el major cost d'aquest projecte és el cost de recursos humans, però actualment tots els projectes relacionats amb aquest camp de la intel·ligència artificial tenen uns costos humans molt elevats ja que són projectes d'investigació. Tot i així, aquest projecte s'incorporarà en la plataforma Freeling de codi obert, per tant, està dissenyat per minimitzar els costos ja que tots els projectes d'investigació podran accedir-hi estalviant-se la recerca ja feta.

Respecte el pla de viabilitat podem dir que és elevat ja que no es el primer projecte d'investigació en aquest àmbit i la planificació mostrada en apartats anteriors garanteix la finalització del projecte.

9.3 Dimensió Social

Respecte l'impacte personal, aquest TFG m'està aportant molts coneixements i metodologies dintre de l'àrea d'intel·ligència artificial, concretament el processament del llenguatge natural.

I respecte l'impacte social, aquest projecte aportarà una nova solució aquest problema que està en investigació, i per tant, aportar una solució a tots aquells projectes que necessiten processar el llenguatge natural de la manera que aquest projecte ofereix. En altres paraules, aquest projecte tindrà impacte sobre altres projectes que necessitin un processament de la relació temporals entre events, tal i com es va explicar en la formulació del problema.

Finalment, el major impacte social que tindrà és que aquesta solució es pujarà a una plataforma de codi obert col·laboratiu per tant estarà a l'abast de tothom agafar-lo, millorar-lo i utilitzar-lo en projectes posteriors.

Implementació

10 Funcionament general del sistema

S'ha implementat un sistema d'aprenentatge automàtic basat en una *Support Vector Machine* que classifica la informació temporal entre dos events.

Aquest sistema rep un document del Freeling, genera els events (Verbs i expressions temporals) i per cada parella d'events genera els features associats.

Un cop generats els features es classifiquen les parelles d'events per determinar la relació temporal. Les relacions temporals poden ser «AFTER» (després), «BEFORE» (abans), «IS_INCLUDED» (està inclòs), «INCLUDES» (inclou), «SIMULTANEOUS» (succeeix alhora), «NONE» (no te relació temporal).

Finalment afegeix les relacions temporals al graf semàntic del Freeling.

11 Generació de Features

Una de les bases fonamentals d'aquests sistema és la generació de features que s'utilitza tant per l'entrenament del model com per la classificació d'events en textos reals.

En aquest projecte s'han definits features a dos nivells, morfosintàctics i semàntics.

11.1 Features morfosintàctics

Els features morfosintàctics que s'han definit són:

- El part of speech de cada event
- La paraula anterior i posterior de cada event
- Totes les paraules entre la parella de events tant la forma en la que apareixen en el text com el seu lemma
- Si algun dels dos events és una data, s'inclou informació extra que proporciona el Freeling
- La distancia en número de paraules que separen els dos events
- La distancia en número de frases
- El número de verbs que hi ha entre els dos events.

Aquests features representen les característiques concretes de cada event, les paraules que els envolten i la relació entre ells en termes de distancia.

11.2 Features semàntics

Els features semàntics que s'han definit són els complements de cada event, si en té:

- Subjecte
- Objecte directe
- Complements circumstancials de temps
- Complements circumstancials de mode

A més s'inclou el *role* que defineix el Freeling.

Aquests features representen el context semàntic al que pertany cada event.

Un exemple dels features generats es pot trobar al Apèndix A.2.

Els features generats s'han de codificar segons el format que accepta la *SVM* en el nostre cas *Libsvm*. La codificació s'ha fet a partir del diccionari que hem generat a la generació de features. Aquest diccionari (Apèndix A.3) ordena els features per freqüència d'aparició.

Veure exemple de features codificats al Apèndix A.4.

12 Entrenament del model

Un cop generats els features i codificats vam definir quins paràmetres exploraríem per tal de fer la cerca del millor model.

Els paràmetres que vam explorar van ser el Kernel, la C, el número de casos «NONE» i la quantitat de features. Aquests paràmetres són aquells que més canvi produeixen en el model i per tant, dintre del nostre temps limitat, vam explorar els que creiem que aportarien més diferencies per trobar el millor model.

- Kernel: Lineal o Quadràtic

- C: rang inicial (0.01, 0,1, 1, 10, 100). També es van fer proves amb 500-1000-10000 en cas de models quadràtics, es comentarà a la avaluació dels resultats el perquè.

- El número de casos «NONE»: En el conjunt de train es generaven 10 vegades més «NONES» que casos positius i per tal d'evitar que el model respongués «NONE» sempre vam variar el número de «NONES» des de 300K a 100K.

- La quantitat de features: La quantitat de features es va controlar definint una freqüència mínima en la qual si el feature apareix menys vegades que aquesta no es codificava i per tant es reduïa el nombre de features. En els experiments vam utilitzar el tall de 5, 10 i 50 com a freqüències mínimes. També es va recodificar el corpus d'entrenament variant els features generats, es donarà més detalls a l'avaluació dels resultats.

Com a última exploració del model vam fer un entrenament canviant l'estratègia del classificador. En lloc de classificar entre les 6 classes directament vam utilitzar la classificació binaria, és a dir, la classificació es fa en dues fases, primer es classifica si la parella d'events té una relació temporal positiva o «NONE» i en cas de ser positiva es torna a classificar entre les 5 restants.

12.1 Avaluació dels resultats

Els tests s'han anomenat segons el següent format «c_#Feats-#Nones-Kernel-C».

#Feats → es el número de features, hi han 3 tipus:

1. 5, 10 o 50 : Aquest número indica la freqüència mínima perquè un feature sigui codificat. La intenció d'aquest valor es aconseguir el millor resultat amb la freqüència mínima més alta.
2. 5/10/50 -- : El doble guió indica que aquesta versió fa swap de les classes positives que estaven codificades a la inversa en els textos etiquetats.
3. 5/10/50-t0 : Aquesta nomenclatura indica que els casos amb t0 s'han obviat ja que en els casos reals mai podrem assumir un t0 com la data global del document.

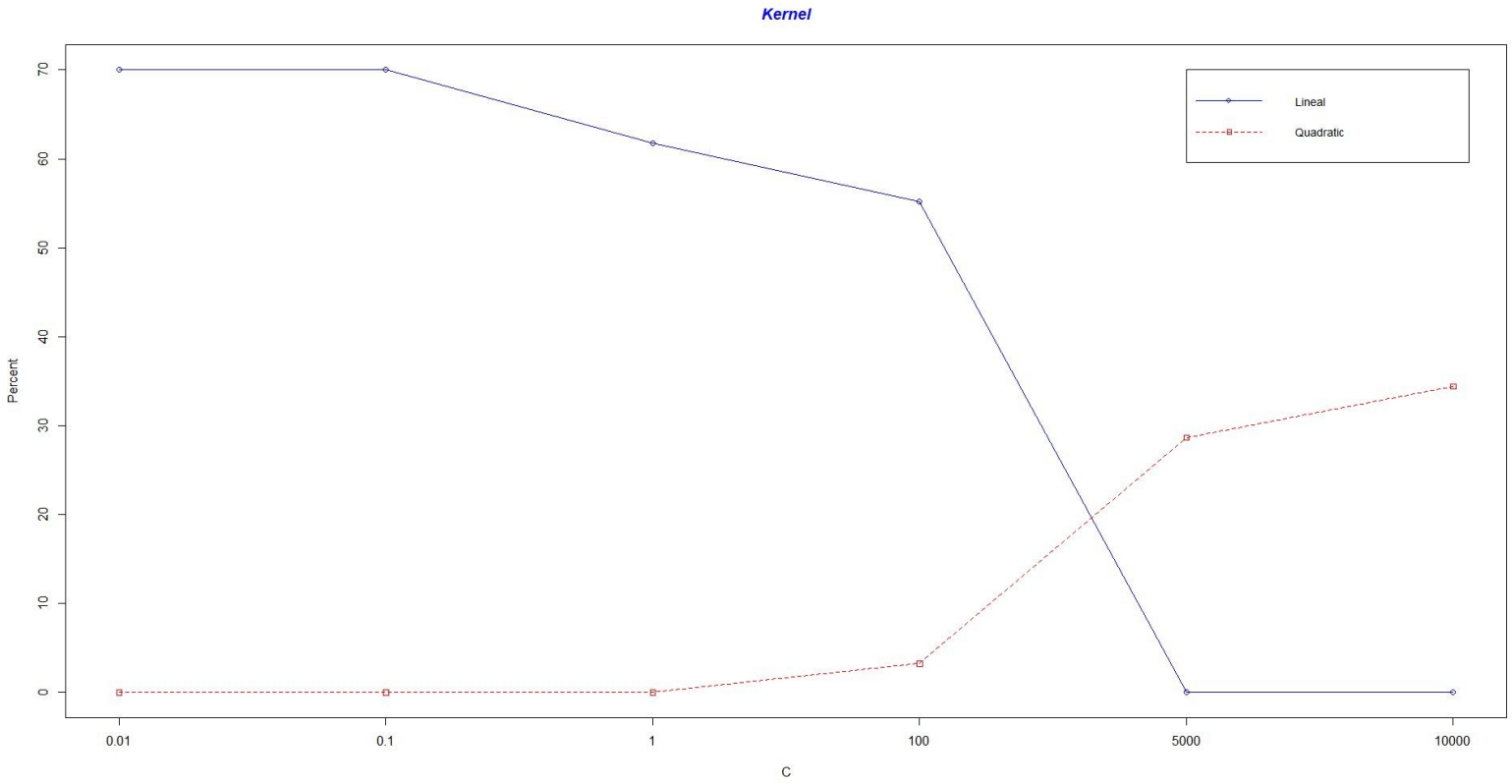
#Nones → es el número de casos negatius que s'ha utilitzat per l'entrenament del model. Els casos positius són 100K.

El Kernel i la C s'han definit en l'apartat anterior.

Primer de tot definir que si F1 són 0 vol dir que el model ha predit sempre «NONE» i encara que així té un accuracy alt ja que la proporció de «NONES» es molt alta no ens interessa ja que volem un model que sigui capaç de reconèixer classes positives no «NONES». Es per això que davant d'aquests models hem posat valors 0.

Per les comparatives mostrarem la F1 de cada test ja que es una mitja entre precisió i recall.

- Kernel



[Gràfica 1]

Les proves que hem fet amb els diferents kernels han donat resultats molt clars. El kernel lineal dóna millors resultats que el quadràtic.

El kernel quadràtic si la C es petita sempre prediu «NONE» i contra més gran sigui la C millor resultat trobem però La F1 més gran aconseguida (34.43) és més baixa que la F1 que aconseguim amb el kernel lineal (70.00).

Per tant vam concloure ràpidament que per aquest problema el kernel lineal es millor.

- C

Tal i com es veu a la gràfica 1, encara que amb el kernel quadràtic contra més gran sigui la C millors resultats, amb el kernel lineal hem obtingut el millor model entre 0.1 i 0.01.

Per tant vam realitzar la resta de proves amb el kernel lineal i aquestes dues Cs.

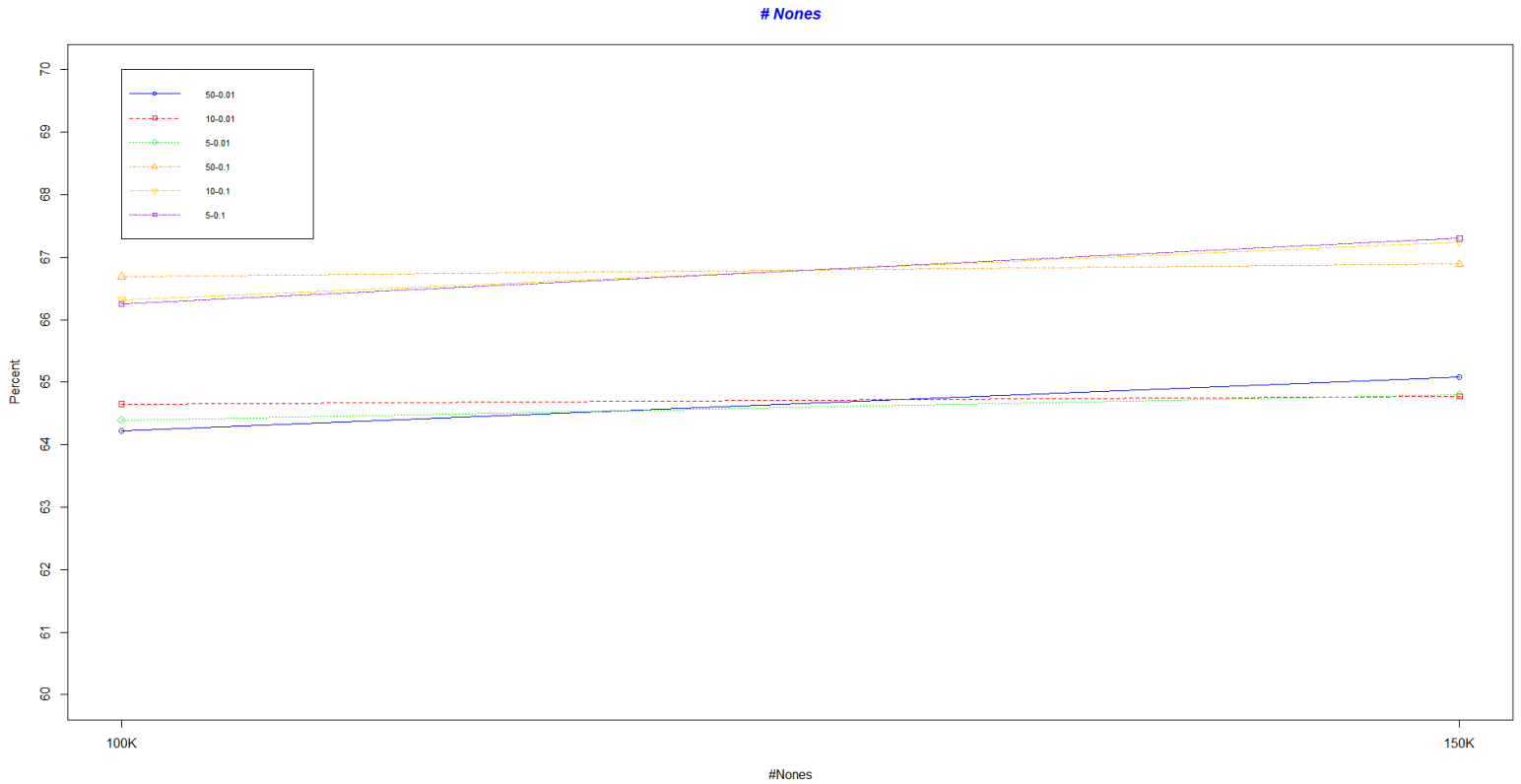
- El número de casos «NONE»

Vam fer experiments amb el número de nones a veure si afegint o traient el model millorava. Contra més número de nones més tendeix el model a predir sempre «NONE» però necessitem que el sistema aprengui en un entorn semblant al real on el número de «NONES» es molt més gran que el número de casos positius.

Les primeres proves les vam fer amb 300K i 200K però els resultats no eren bons i trigava molt l'entrenament es per això que els resultats obtinguts els vam descartar. Vam seguir reduint «NONES» fins a 150K on van començar a aparèixer bons models.

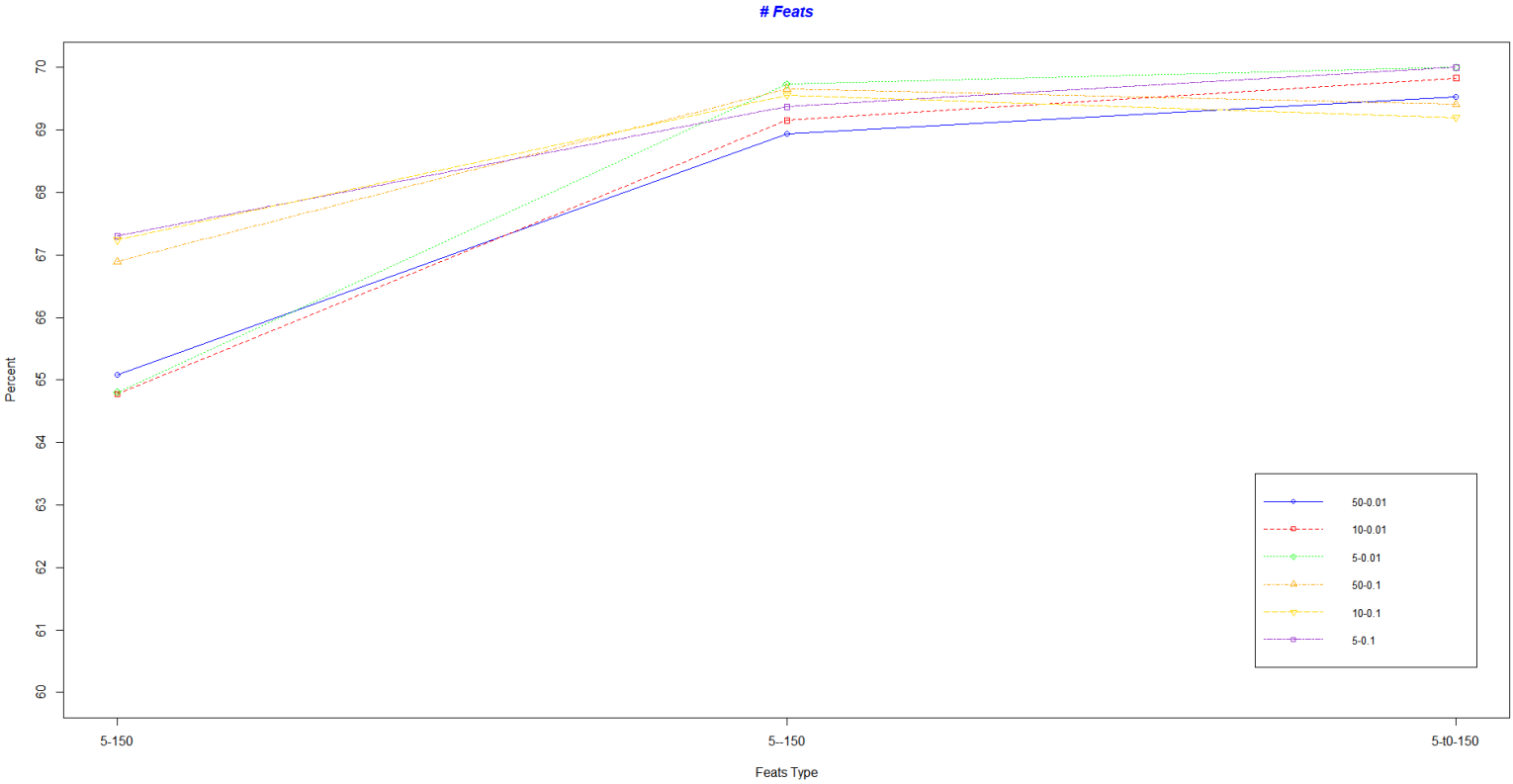
Vam reduir de 150K a 100K (igualant el numero de casos positius) per tal de veure si així aconseguíem un millor model.

Els resultats, tal i com es veu a la gràfica 2, van ser que amb 100K «NONES» el model era una mica pitjor encara que no gaire. I com estàvem al voltant de 67% d'F1 vam decidir que no valia la pena baixar a 100K, per tant vam seguir investigant amb 150K nones.



[Gràfica 2]

- La quantitat de features



[Gràfica 3]

Per una banda, respecte el número de features, veiem que els talls de diccionari estan molt igualats, és a dir, tant models amb 5, 10 o 50 tenen un resultat similar tot i així el 5, majoritàriament, han estat els models amb millors resultats. Per tant, tot i que amb un tall 5 el número de features augmenta el guany amb tall 10 no es suficient.

Per altra banda, tenim 3 tipus de models en la gràfica 3: 5-150, 5--150 i 5-t0-150.
La diferencia d'aquests models són la manera de generar features. En la primera (el primer conjunt de proves de la gràfica 3) codificava tot el corpus d'entrenament tal i com estava etiquetat però vam veure que hi havien alguns casos positius que estaven etiquetats al revés del ordre d'aparició en el text i per tant al codificar-los estàvem generant malament alguns casos positius. Llavors vam tornar a codificar el corpus d'entrenament fent swap de les classes codificades al revés i vam millorar fins al 69% (segon conjunt de proves de la gràfica 3).

Per últim, vam veure que en el corpus d'entrenament estava etiquetat la expressió temporal t0 com la data de la noticia i per tant tots els events tenien una relació amb aquesta data, però això mai passarà en els casos reals ja que no tenim manera d'assumir que una data sigui global com ho es el t0 en el corpus d'entrenament. Per tant vam fer una prova eliminant aquelles parelles que identificaven una relació amb t0 i vam aconseguir una petita millora fins al 70% (tercer conjunt de proves de la gràfica 3).

12.2 Taula dels millors tests:

Test	Precision	Recall	F1
c_50-150-0-0.1	65.93	67.87	66.89
c_10-150-0-0.1	65.45	69.13	67.24
c_5-150-0-0.1	66.02	68.63	67.30
c_50--150-0-0.01	69.96	67.96	68.94
c_10--150-0-0.01	70.29	68.04	69.15
c_10-t0-150-0-0.1	66.69	71.91	69.20
c_5--150-0-0.1	66.93	71.99	69.37
c_50-t0-150-0-0.1	67.38	71.57	69.41
c_50-t0-150-0-0.01	71.09	68.04	69.53
c_10--150-0-0.1	67.14	72.16	69.56
c_50--150-0-0.1	67.62	71.83	69.66
c_5--150-0-0.01	70.51	68.97	69.73
c_10-t0-150-0-0.01	70.46	69.22	69.83
c_5-t0-150-0-0.1	67.31	72.92	70.00
c_5-t0-150-0-0.01	70.63	69.39	70.00

12.3 Model binari

- Kernel Quadràtic

Test Name	Precision	Recall	F1
c_5-b-150-1-0.1	0.00	0.00	0.00
c_5-b-150-1-1	37.52	93.19	53.50
c_5-b-150-1-100	39.92	95.04	56.22

Test Name	Precision	Recall	F1
c_5-p-150-1-0.1	42.98	42.98	42.98
c_5-p-150-1-1	42.98	42.98	42.98
c_5-p-150-1-100	43.06	43.06	43.06

- Kernel Lineal

Test Name	Precision	Recall	F1
c_5-b-300-0-0.01	85.18	61.40	71.36
c_5-b-300-0-0.1	80.21	69.22	74.31
c_5-b-150-0-0.01	71.63	77.71	74.55
c_10-b-150-0-0.01	71.43	78.22	74.67
c_10-b-150-0-0.1	69.80	80.66	74.83
c_5-b-150-0-0.1	69.73	81.16	75.01

Test Name	Precision	Recall	F1
c_10-p-150-0-0.01	81.24	81.24	81.24
c_5-p-150-0-0.01	81.41	81.41	81.41
c_5-p-300-0-0.01	81.41	81.41	81.41
c_5-p-150-0-0.1	83.94	83.94	83.94
c_5-p-300-0-0.1	83.94	83.94	83.94
c_10-p-150-0-0.1	84.02	84.02	84.02

Els tests amb 'b' són aquells models que classifiquen entre relació positiva o «NONE» i els tests amb 'p' són aquells models que classifiquen entre les 5 classes positives.

S'han realitzat proves amb el kernel quadràtic per verificar si el comportament era el mateix que amb els models no binaris. I tal com veiem a les taules el seu rendiment es inferior al dels models amb kernel lineals.

Sobre les proves amb el kernel lineal podem veure que les proves amb 150K «NONES» funciona millor que amb 300K «NONES» i que la millor C es 0.1.

El càlcul sobre la qualitat del model binari s'ha de fer amb l'error acumulat de les dues classificacions. Per tant, en les proves realitzades trobem que la F1 amb l'error acumulat estarà entre 50%-60%, resultat pitjor que el millor model trobat amb el classificador de 6 classes.

Donat que el resultat no s'ha aconseguit millorar amb aquesta estratègia de classificació es va optar per utilitzar el millor model de 6 classes.

13 Mòdul del Freeling

Un cop trobat el millor model es va implementar un mòdul que, mitjançant el model anterior i el libsvm, classifiques un document del Freeling segons la relació temporal entre dos events.

Un cop generades aquestes relacions s'afegeixen al semàntic graf del Freeling per tal de poder mostrar la nova informació.

Per tal d'afegir les relacions al semàntic graf, s'ha creat la entitat «Relation_Temporal» que defineix una relació temporal entre dos events. Aquestes entitats representen nova informació a representar.

Un XML d'exemple es pot veure a l'Apèndix A.5.

La API d'aquest mòdul està formada per dos noves classes. Una genera les parelles d'events i els seus features associats i l'altra configura el classificador amb el model i extreu les relacions temporals de les parelles d'events.

13.1 Generador de events i features:

```
class featGenerator
{
public:
    list<event> events;
    map<string, int> dic;
    const string numericFeatures[3] = {"nWord","nSen","nVerb"};
    list< pair<string, string> > conexions;
    bool lexicalfeats, syntactfeats;
    ofstream out[2];
    string current_features;


    featGenerator();
    featGenerator(bool lexicalfeats, bool syntactfeats);
    ~featGenerator();

    //Add elements to structures
    void createEvents(const list<paragraph::const_iterator> &ls);
    void addEvent(event e);
    void addFeatToDic(string feat);
    void addConexion(string ei, string ej);
    bool openFile(int i, string path);

    //Print structures to control
    void printEvents();
    void printConexions();

    //Principal functions, generate features and dicctionary
    void printDic(int numOfClasses);
    void printFeat(string feat);
    void printFeatsSet(set<string> feats);
    void generateFeatures(const list<paragraph::const_iterator> &ls);
    void generateLexicalFeats(event &ei, event &ej, const list<paragraph::const_iterator> &ls);
    void generateSyntacticalFeats(event &ei, event &ej, const list<paragraph::const_iterator> &ls);
    string generateFeatures2String(event &ei, event &ej, const list<paragraph::const_iterator> &ls);
    list<pair<int,int>> codeFeatures(list<string> &features, map<string,int> &dic);

    //gets
    list<event> getEvents();
    list<std::pair<event,event>> getPairs();
    string getCurrentFeatures();


    //sets
    void setBooleans(bool lexicalfeats, bool syntactfeats);
    void resetCurrentFeatures();

private:
    string trataNumeric(string f);
    void printDateInfo(string dateInfo, string word);

    static bool sortFunc(pair<string,int> first, pair<string,int> second);
    list<string> split(string s, char delim);
};
```

Aquesta classe et permet generar events i codificar els features associats, tant imprimint-los en un fitxer com retornant-los en un string.

13.2 Classificador de relacions:

```
class relationclassificator
{
 private:
    /// classifier
    classifier *classif;

    ///dictionary
    map<string,int> dic;

  public:
    /// Constructor
    relationclassificator(const std::wstring &);
    /// Destructor
    ~relationclassificator();

    /// predict from string
    void predict(const freeling::document &doc);

  private:
    list<string> split(string s, char delim);
    wstring string2wstring(string s);
};
```

Aquesta classe configura un classificador, en les nostres proves hem utilitzat la *libsvm* però es podria utilitzar un altre, i un diccionari de features que conté la traducció del feature codificat al feature en mode text.

Aquesta classe té una funció *predict* que donat un document del Freeling executa tota la funcionalitat del sistema i afegeix la nova informació al *semantic graph* del document.

14 Tests Finals

Un cop generat el model i implementat el mòdul del Freeling s'han realitzat uns tests simulant casos reals i utilitzant el conjunt de test que ofereix TempEval-3 per tal de verificar si el nostre entrenament entre el conjunt de train i development manté els seus resultats o baixen.

El conjunt de test que ofereix el TempEval-3 es un conjunt de 20 noticies etiquetades com el conjunt de train. En aquest projecte s'han generat conjunts de development amb la mateixa mida que el de test.

14.1 Avaluació dels resultats

Els resultats que hem obtingut han sigut els següents:

Test Name	Precision	Recall	F1
c_10test-150-1-10000	22.81	9.57	13.48
c_test-5-t0-150-0-0.01	34.19	29.45	31.64

Els resultats amb el conjunt de test que ens proporciona el concurs TempEval-3 té uns resultats molt més baixos que les proves amb el conjunt de development, tant el millor model amb kernel lineal com quadràtic.

Per tal de verificar les nostres proves amb el conjunt de development vam crear dos conjunts nous:

Test Name	Precision	Recall	F1
c_5-dev2-t0-150-0-0.01	73.44	64.89	68.90
c_5-dev-t0-150-0-0.01	75.51	68.90	72.05

Els resultats es mantenien sobre el 70%.

14.2 Perquè obtenim aquests resultats de test?

L'explicació de perquè disminueix tant l'F1 amb el conjunt de test es deu al *overfitting*. [14] Aquest overfitting es degut a que el conjunt de train conté exemples de dos diaris i per això s'ha sobreajustat el sistema al format d'aquests diaris.

Aquest fet es pot contrastar amb el conjunt de test que conté molta més varietat de diaris i aquestes mostres són suficientment diferents del conjunt de train perquè els resultats dels participants del concurs TempEval-3 no van ser considerablement millors al 30%. [2]

Participant	F1	P	R
ClearTK-2	36,26	37,32	35,25
ClearTK-4	35,86	35,17	36,57
ClearTK-1	35,19	37,64	33,04
UTTime-5	34,9	35,94	33,92
ClearTK-3	34,13	33,27	35,03
El Nostre projecte	31.64	34.19	29.45
NavyTime-1	31,06	35,48	27,62
UTTime-4	28,81	37,41	23,43
JU-CSE	26,41	21,04	35,47
NavyTime-2	25,84	31,1	22,1
KUL-TE3RunABC	24,83	23,35	26,52
UTTime-1	24,65	15,18	65,64
UTTime-3	24,28	15,1	61,99
UTTime-2	24,05	14,8	64,2

15 Conclusions i treball futur

L'objectiu principal d'aquest projecte ha estat implementar un sistema capaç de classificar una parella d'events segons la seva relació temporal. Podem concloure que l'objectiu principal s'ha assolit. El sistema es capaç de rebre un document del Freeling i classificar les parelles d'events segons la seva relació temporal mostrant aquesta informació en el graf semàntic. La resta d'objectius definits a la planificació del projecte també s'han assolit.

La generació de features s'ha implementat tant per la codificació del dataset d'entrenament com per la codificació del text d'entrada del sistema. Es podria millorar fent una investigació més profunda sobre les millores del model segons nous conjunts de features. En aquest projecte s'ha utilitzat el conjunt que millors resultats han donat però es poden definir altres conjunts diferents.

L'entrenament del model s'ha fet mitjançant una petita investigació variant els paràmetres que hem considerat més rellevants per una *Support Vector Machine* , el kernel, la C, el número de casos negatius i el número de features. L'entrenament es podria millorar ampliant la ramificació de paràmetres entrenats i aplicant els nous conjunts de features.

El mòdul del Freeling proporciona una estructura simple que implementa el sistema des de l'input d'un document del Freeling fins la incorporació de les relacions temporals al graf semàntic. El mòdul es podria ampliar afegint un post procés que elimines les incoherències entre relacions, és a dir, que cap relació contradigui cap altra.

En resum, l'objectiu principal d'aquest projecte s'ha assolit creant una base simple i fàcilment ampliable que permet futures millores del sistema. Els resultats obtinguts han entrat dintre de la mitja de resultats obtinguts pels participants del concurs TempEval-3 tot i que com a resultats globals l'overfitting es gran.

16 Bibliografia

[1] Xavier Carreras and Isaac Chao and Lluís Padró and Muntsa Padró. ***FreeLing: An Open-Source Suite of Language Analyzers.*** *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04), 2004*.
http://nlp.lsi.upc.edu/publications/papers/carreras04.pdf
Freeling website http://nlp.lsi.upc.edu/freeling/

[2] Naushad UzZaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen and James Pustejovsky. **SemEval-2013 Task 1: TEMPEVAL-3: Evaluating Time Expressions, Events, and Temporal Relations.**
http://www.aclweb.org/anthology/S/S13/S13-2001.pdf
TempEval-3 website https://www.cs.york.ac.uk/semeval-2013/task1/

[3] Nathanael Chambers and Shan Wang and Dan Jurafsky. ***Classifying Temporal Relations Between Events*.**
http://nlp.stanford.edu/pubs/acl07-chambers.pdf

[4] William F. Styler IV1 , Steven Bethard2 , Sean Finan3 , Martha Palmer1 , Sameer Pradhan3 , Piet C de Groen4 , Brad Erickson4 , Timothy Miller3 , Chen Lin3 , Guergana Savova3 and James Pustejovsky5. ***Temporal Annotation in the Clinical Domain*.**
https://transacl.org/ojs/index.php/tacl/article/viewFile/305/40

[5] Marta Tatu and Munirathnam Srikanth. ***Experiments with Reasoning for Temporal Relations between Events*.**
http://www.aclweb.org/anthology/C08-1108

[6] Paramita Mirza. ***Extracting Temporal and Causal Relations between Events*.**
http://www.aclweb.org/anthology/P14-3002

[7] Leon R.A. Derczynski. ***Determining the Types of Temporal Relations in Discourse*.**
http://etheses.whiterose.ac.uk/4068/1/phdthesis.pdf

[8]Open Source Software Pugixml, Maintained by Arseny Kapoulkine.
http://pugixml.org/

[9]Chih-Chung Chang and Chih-Jen Lin. ***LIBSVM--A Library for Support Vector Machines*.**
https://www.csie.ntu.edu.tw/~cjlin/libsvm/

[10]Enciclopedía lliure wikipedia.
https://ca.wikipedia.org/wiki/M%C3%A0quina_de_vector_de_suport

[11] Enciclopedía lliure wikipedia.
https://ca.wikipedia.org/wiki/Processament_de_llenguatge_natural

[12] Tarifa d'internet
https://www.jazztel.com/

[13] Tarifes Transport Metropolità de Barcelona
https://www.tmb.cat/es/barcelona/tarifas-metro-bus/abonos/t-jove#hdr_preus_zona

[14] Enciclopedía lliure wikipedia.
https://es.wikipedia.org/wiki/Sobreajuste

Apèndix A

Exemples de inputs i outputs generats en el projecte

En aquest apèndix es mostren exemples de fitxers utilitzats pel sistema que s'ha desenvolupat en aquest projecte. Tant fitxers d'input utilitzats en l'entrenament del model com els de resultat.

A.1 Text etiquetat TempEval-3

```
−<TimeML xsi:noNamespaceSchemaLocation="http://timeml.org/timeMLdocs/TimeML_1.2.1.xsd">
  <DOCID>AFP_ENG_19970409.0021</DOCID>
 −<DCT>
    SINGAPORE,
    <TIMEX3 tid="t0" type="TIME" value="1997-04-09" temporalFunction="false" functionInDocument="CREATION_TIME">April 9 , 1997</TIMEX3>
    (AFP)
  </DCT>
  <TITLE>Noon rubber prices</TITLE>
 −<TEXT>
    <TIMEX3 type="DATE" value="1997-04-09" tid="t1">Wednesday</TIMEX3>
    's noon rubber prices in Singapore cents per kilo
    <EVENT class="OCCURRENCE" eid="e1">provided</EVENT>
    by the Singapore Commodity Exchange: Buyers Sellers Int 2 RSS May 169.00 170.00 Nominal (N) Int 3 RSS
    <TIMEX3 type="DATE" value="1997-04-09" tid="t3">May 169.00 170.00 N</TIMEX3>
    <TIMEX3 type="DATE" value="1997-04-09T04:00" tid="t4">Int 4 RSS May 163.75 164.75 N</TIMEX3>
    <TIMEX3 type="DATE" value="1997-04-09T05:00" tid="t5">Int 5 RSS May 160.00 161.00 N</TIMEX3>
    <TIMEX3 type="DATE" value="1997-04-09T01:00" tid="t6">No. 1</TIMEX3>
    Air Dried Sheet 189.00 191.00 N Int 1X Thin Pale Crepe 234.50 238.50 N Int 1 Thin Pale Crepe 228.50 232.50 N Int 2 Thin Pale Crepe 220.50 222.50 N Int 3 Thin Pale Crepe 215.50 217.50 N Int 2 Thin Brown Crepe 132.00 134.00 N Int 3 Thin Brown Crepe 130.00 132.00 N Int 4 Thin Brown Crepe 128.00 130.00 N
    SSR/SMR 20 May 163.00 165.00 N
   −<TIMEX3 type="DATE" value="1997-04-09" tid="t8">
      June 164.00 166.00 N SSR 50 May 161.00 163.00 N
    </TIMEX3>
    <TIMEX3 type="DATE" value="1997-04-09" tid="t9">June 162.00 164.00 N</TIMEX3>
  </TEXT>
  <MAKEINSTANCE eiid="ei1" eventID="e1" pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
  <TLINK lid="l1" eventInstanceID="ei1" relatedToTime="t0" relType="BEFORE"/>
  <TLINK lid="l2" eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"/>
</TimeML>
```

A.2 Features generats

A continuació es mostra els features associats a una parella d'events

AFTER w1Before=, w1After=(w2Before=prices w2After=1.1 w1form=April_1_,_1997 w2form=dipped w1Lemma=[??:1/4/1997:??.??:??] w2lemma=dip WIMLemma=(WIMLemma=) WIMLemma=1.1 WIMLemma=[??:1/4/1997:??.??:??] WIMLemma=[??:??/??/??:12.00:pm] WIMLemma=afp WIMLemma=at WIMLemma=dip WIMLemma=fall WIMLemma=malaysian WIMLemma=market WIMLemma=percent WIMLemma=price WIMLemma=share wordInMiddle=(wordInMiddle=) wordInMiddle=1.1 wordInMiddle=AFP wordInMiddle=April_1_,_1997 wordInMiddle=Malaysian wordInMiddle=at wordInMiddle=dipped wordInMiddle=falls wordInMiddle=market wordInMiddle=midday wordInMiddle=percent wordInMiddle=prices wordInMiddle=share w1dayOfMonth=1 w1Month=4 w1Year=1997 nWord=16 nVerb=2 nSen=1 w2Sense_01577093-v w2OBJ_percent

A.3 Diccionari

Aquest diccionari s'ha generat amb la generació de features i s'utilitza per codificar-los

6 nWord 375563
7 nVerb 375563
8 nSen 375563
9 WIMLemma=the 271640
10 wordInMiddle=the 266466
11 wordInMiddle=, 250860
12 WIMLemma=, 250860
13 WIMLemma=to 219953
14 wordInMiddle=to 219478
15 wordInMiddle=. 210776
16 WIMLemma=. 210776
17 WIMLemma=be 197543
18 WIMLemma=" 194485
19 wordInMiddle=" 194485
20 WIMLemma=of 184894
[...]
237 w1OBJ_be 17600
238 wordInMiddle=only 17566
239 wordInMiddle=last 17472
240 WIMLemma=win 17452
241 wordInMiddle=work 17250
242 w2Before=, 17242
[...]
329 WIMLemma=effort 13741
330 WIMLemma=kill 13631
331 WIMLemma=president 13556
332 w2Before=the 13536
333 wordInMiddle=think 13456
334 WIMLemma=[L:??/??/??:??.??:??] 13386

A.4 Features Codificats

A continuació es mostren els features del apèndix A.1 codificats amb el diccionari de l'apèndix A.2

1 6:16 7:2 8:1 37:1 57:1 63:1 65:1 66:1 67:1 68:1 276:1 279:1 280:1 405:1 406:1 484:1 488:1 489:1 591:1 641:1 741:1 826:1 841:1 2908:1 5128:1 5446:1 8372:1 10038:1 10042:1 11275:1 11708:1 11714:1 12319:1 13713:1 14168:1 14234:1 15217:1 19828:1 19941:1 19962:1 57530:1 64833:1 66670:1

A.5 XML graf semàntic

A continuació es mostra un exemple de la nova informació afegida al graf semàntic del freeling en format XML

```
<frame id="F18" token="t2.33" lemma="oppose.01" sense="10379620-n" >
      <argument role="AM-MNR" entity="W25" />
      <synonym lemma="opponent"/>
      <synonym lemma="opposite"/>
      <synonym lemma="opposition"/>
      <URI knowledgeBase="WordNet" URI="http://wordnet-rdf.princeton.edu/wn30/10379620-n"/>
      <URI knowledgeBase="OpenCYC" URI="http://sw.opencyc.org/concept/Mx8NhB4rvcD6KJwpEbGdrcN5Y29ycB4rv
      <URI knowledgeBase="SUMO" URI="http://ontologyportal.org/SUMO.owl#SocialRole"/>
   </frame>
   <frame id="F19" token="t2.35" lemma="rout.00" sense="01104248-v" >
      <argument role="A1" entity="W26" />
      <synonym lemma="rout"/>
      <synonym lemma="spreadeagle"/>
      <synonym lemma="spread-eagle"/>
      <URI knowledgeBase="WordNet" URI="http://wordnet-rdf.princeton.edu/wn30/01104248-v"/>
      <URI knowledgeBase="SUMO" URI="http://ontologyportal.org/SUMO.owl#Contest"/>
   </frame>
 <relation_tmp id="RT0" w1="F5" w2="F7" relation="AFTER" >
 </relation_tmp>
 <relation_tmp id="RT1" w1="F5" w2="F8" relation="AFTER" >
 </relation_tmp>
 <relation_tmp id="RT2" w1="F7" w2="F8" relation="INCLUDES" >
 </relation_tmp>
 <relation_tmp id="RT3" w1="F15" w2="W21" relation="INCLUDES" >
 </relation_tmp>
 <relation_tmp id="RT4" w1="F15" w2="F17" relation="AFTER" >
 </relation_tmp>
 <relation_tmp id="RT5" w1="F17" w2="F19" relation="INCLUDES" >
 </relation_tmp>
</semantic_graph>
</document>
```

Es pot veure les relacions entre *Frames* (relacions event-event) i relacions entre *Frames* i *Entities* (relacions event-timeexpresion).

Apèndix B

Codi font del projecte

https://github.com/TekuDruida/tfg