Implementació d'un mòdul descodificador per un sistema OCR

Guillem Jardí Geijo

Resum— Els sistemes OCR, sistemes informàtics capaços d'interpretar paraules en una imatge, tenen un gran impacte i possibilitats d'ús en la societat d'avui dia. En aquest projecte es presenta com s'han desenvolupat una sèrie de models amb l'objectiu d'interpretar la lectura de paraules en imatges naturals obtingudes per un sistema OCR. Les característiques que ens proporcionen els grafs, juntament amb la correcta aplicació d'algoritmes de cerca de camins òptims són les principals estratègies utilitzades per al desenvolupament dels tres models implementats.

Paraules clau— OCR, Graf, Vèrtex, Camí òptim, Histogrames, Descodificar, Paraules, Letres, Vector característic.

Abstract— OCR systems, computer systems capable to extract words from an image, have a great impact and possibilities of use in today's society. This project shows how a series of models have been developed with the aim of interpreting the reading of words in natural images obtained by this system. The characteristics that the graphs provide to us, with the correct application of optimal path search algorithms are the main strategies used for the development of the three models implemented in this project.

Index Terms— OCR, Graph, Vertex, Optimal path, Histograms. Decode, Words, Letter, Characteristic vector.

1 Introducció

AQUEST projecte neix com una iteració del projecte "Word Spotting and Recognition with Embedded Attri-butes" [1], aquest tenia com a objectiu el reconaixement de paraules en imatges naturals. Mitjançant la utilització de tècniques de visió per computador i una xarxa neuronal aconseguien un etiquetatge ajustat de les paraules i posteriorment, amb l'ajuda d'un gran diccionari, podien determinar quines paraules hi apareixen en una determinada imatge amb un alt percentatge d'encert.

L'etiquetatge de les paraules utilitzat consisteix en un histograma probabilístic de 604 dimensions. Aquest histograma està format per un conjunt de 14 histogrames de 36 dimensions on cada una d'aquestes dimensions representa un possible caràcter de l'alfabet anglès (de la "a" a la "z" i de "0" a "9"). Els 14 histogrames són definits per la utilització de la següent metodologia; donada una paraula, aquesta es divideix en dues regions de grandària equivalent creant dos histogrames representatius per a cada una de les regions. Seguidament, la paraula és dividida en tres regions, i novament es crea un histograma de 36 dimensions representatiu per a cada regió de la paraula, així successivament fins a 5 divisions de la paraula. Per tant, 504 dimensions de l'etiquetatge d'una paraula estan per l'estratègia esmentada (2+3+4+5)*36. A més a més, s'afegeixen els 50 digrames més comuns del segon ni-vell, afegint les 100 dimensions restants i constituint l'histograma de 604 dimensions per a l'etiquetatge de la paraula.

En l'inici de la introducció s'ha esmentat la necessitat de la utilització d'un gran diccionari amb milers de paraules etiquetades per tal de poder determinar la paraula o paraules que apareixen en una imatge. Aquesta estratègia té un gran inconvenient i és que el sistema no és capaç d'interpretar i determinar les paraules d'una imatge correctament si aquestes no es troben en el diccionari. És en aquest punt on el projecte cobra sentit, tenint com a objectiu el desenvolupament d'un model capaç de descodificar els histogrames en la paraula que representa, sense la necessitat de la utilització d'un diccionari per aquesta finalitat, mitjançant tècniques de grafs i de cerca de camins òptims [2].

S'han desenvolupat fins a tres models diferents per a la resolució del problema plantejat. Cadascun dels models neix com una iteració del model anterior amb l'objectiu de garantir uns millors resultats en la descodificació, així com una major optimització en els algoritmes desenvolupats. En els diferents models s'hi destaquen dues fases ben diferenciades, la primera, centrada en la construcció d'un graf característic que representa l'estructura de la paraula a descodificar, aquest s'obté de la interpretació del vector característic. I una segona fase centrada en la cerca del camí més òptim d'aquest, donant com a resultat la paraula més òptima representada pel graf.

- E-mail de contacte: guillem.jardi@e-vampus.uab.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Ernest Valveny (Ciències Computació)
- Curs 2018/19

En les pròximes seccions de l'article es presenten les característiques dels tres models implementats, així com els diferents reaultats que s'han obtingut per cadescun d'ells. Al llarg de la resolució i avalació dels diferents models s'han utilitzat histogrames resultants de la codificació PHOC [1] (histograma característic de la paraula), per a paraules obtingudes de la interpretació d'imatges, així com, histogrames resultants de la codificació de *strings* literals. En el primer cas, els histogrames que s'obtenen són histogrames probabilístics, i en el segon cas, obtenim histogrames binaris.

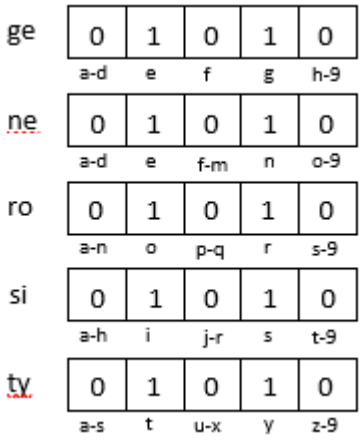
2 IMPLEMENTACIÓNS DELS MODELS

En aquesta secció es presenta com s'han anat implementat els diferents models desenvolupats, així com les principals característiques que presenten, tant en la construcció del graf com en l'obtenció del camí òptim d'aquest. Juntament amb els diferents resultats obtinguts per cadascun d'ells.

El llenguatge de programació utilitzat es Python [3] en la seva versió 2.7.15, l'elecció d'aquest llenguatge ha estat perquè permet un molt bon maneig d'estructura de dades de tipus llista i diccionaris, amb una fàcil implementació i optimització. D'altra banda també permet una programació orientada a objectes, molt necessària per definir l'estructura de dades d'un graf.

2.1 Histogrames binaris

Els primers models representatius desenvolupats per a la resolució del problema, s'han centrat en la interpretació i descodificació de les paraules partint d'un histograma característic binari, obtingut d'una representació exacte. Mitjançant un diccionari de més de 80.000 paraules, s'han codificat totes elles amb la tècnica PHOC, per tal d'obtenir uns histogrames sense soroll, on els valores del mateix son 0 o 1 en funció de si la lletra hi està representada o no en la paraula a descodificar. Els histogrames utilitzats han estat els histogrames del 5è nivell, amb una representació de 36 caràcters, que van de la "a" a la "z" i del "0" al "9", on cadascú representa una regió diferent de la paraula, evitant la diferenciació entre lletres majúscules i lletres minúscules. Així docs, obtenim un histograma de 180 dimensions determinat per 5 histogrames de 36 dimensions. En la Il·lustració 1 es presenta com és l'histograma del 5è nivell de la paraula "generosity".



Il·lustració 1. Histograma del 5é nivell per a la paraula "generosity"

2.2 Model 1: Optimitat per regions.

El primer model desenvolupat segueix una estratègia de la construcció del graf i de la cerca del camí òptim molt arrelada a la clara definició de les regions de la paraula representades en l'histograma característic. La representació del graf ve donada per la definició d'un graf per cadescun dels nivells i la la solució óptima consisteix en la unió dels camins òptims de cadascuna de les regions. A continuació s'indica l'estructura de dades del model així com les principals carecterísticas d'implementació dels procediments més rellevants.

2.2.1 Optimitat per regions: estructura de dades.

Per tal de garantir una correcta definició i representació del graf, així com la possibilitat d'aplicar diferents mecanismes per a la cerca del camí òptim, s'ha definit una estructura de dades amb dues entitats clarament diferenciades, l'entitat Vèrtex i l'entitat Graf.

En l'entitat vèrtex es defineixen les característiques necessàries per a definir un vèrtex d'un graf dirigit ponderat, on cadascun d'aquests representa una de les lletres de la paraula a descodificar. Per altra banda, a causa de les característiques del problema, cadascuna de les lletres representades en l'histograma característic, té associada una probabilitat, aquesta probabilitat defineix la possibilitat que la lletra estigui, o no, representada en la paraula. Sí ve és cert que en aquestes primeres iteracions, on treballem amb histogrames binaris, la probabilitat definida no és rellevant. En les posteriors iteracions, on l'histograma a descodificar està definit per probabilitats, consistirà en una de les principals característiques que determinarà la paraula descodificada resultant. D'altra banda, en la classe Vèrtex, també es defineixen les connexions del Graf, així doncs, en cadascun dels vèrtexs s'indica els seus vèrtexs veïns juntament amb una puntuació que indica el grau d'optimització de la unió.

L'altra classe desenvolupada, la classe Graf, té com a objectiu la definició de l'estructura del graf característic, juntament amb l'estratègia de cerca del camí òptim que dóna com a resultat la descodificació de la paraula. Tal com s'ha mencionat anteriorment, en aquest primer model desenvolupat, l'estructura del graf ve donada per diferents grafs representatius per cadascun dels subconjunts de la paraula. Aquests grafs, no tenen connexió entre ells, per tant, es creen cinc subgarfs no connexos on el camí òptim està format per la unió dels resultats obtinguts en cadascuna de les cerques dels subgrafs. Tal com s'ha mencionat anteriorment, les diferents connexions que hi ha en el graf, es mantenen i es gestiona des de l'entitat vèrtex, aquest fet implica, que en la classe graf, únicament es guardin, en una estructura de llista, els diferents vèrtexs que formen els subgrafs de la paraula. En la Il·lustració 2, es presenta un exemple dels vèrtexs creats

per a la paraula "generosity" i com es guardarien en l'estructura del graf definida.

L'objecte Graf, no únicament té l'objectiu d'albergar l'estructura del graf, sinó que també incorpora els procediments i mètodes necessaris per a la cerca del camí òptim, que dóna com a resultat la descodificació de la paraula. El camí òptim d'un subconjunt de la paraula es defineix pel càlcul de tots els camins possibles a partir d'un vèrtex del subconjunt. Aquests camins representen l'ordre que menten les lletres dins del subconjunt. Un cop estudiades totes les combinacions possibles, la combinació guanyadora determinarà l'orde de les lletres en aquella regió de la paraula. En acabar l'estudi per les cinc regions, s'obté l'ordenació de les lletres més òptima de l'histograma descodificat.

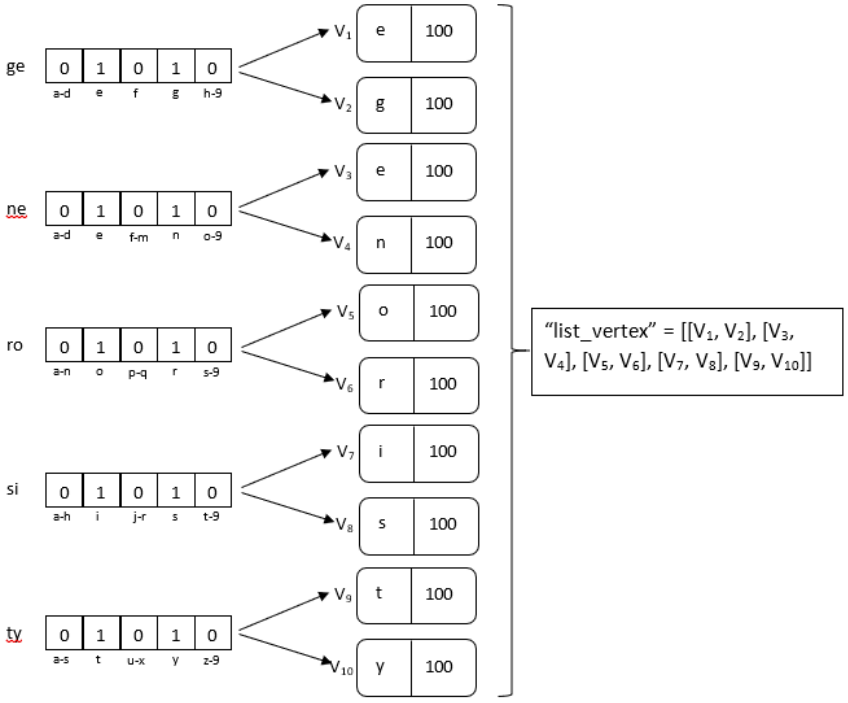
2.2.2 Optimitat per regions: creació del graf.

En aquesta secció s'explicarà l'estratègia utilitzada per a la creació del graf en la primera implementació. Tal com s'ha esmentat anteriorment, la representació del graf és definida per la interpretació de l'histograma del 5è nivell de la paraula. Aquest histograma és de 180 caràcters, ja que conté les característiques de 5 subconjunts de lletres de grandària similars i creats d'esquerra a dreta, cadascú d'aquests subconjunts és representat per un histograma de 36 caràcters. No obstant això, per a paraules d'una i dues lletres, l'histograma utilitzat és el del primer i segon nivell respectivament. En aquests casos, les paraules no tenen representació en els histogrames del 5è nivell.

A partir de l'histograma seleccionat, s'extrauen les possibles lletres que hi ha a cada subconjunt de la paraula. Aquesta informació és introduïda en un diccionari on les claus són els possibles caràcters i el valor la seva probabilitat. Recordem que en aquesta primera iteració la probabilitat sempre serà del 100% i, per tant, més aviat ens indica quants caràcters d'aquell tipus hi ha en el subconjunt de la paraula. Ja que un caràcter que hi aparegui dues vegades en el subconjunt es veurà representat per una probabilitat del 200%. Aquest diccionari es crea per a cada regió de la paraula i són guardats en una llista.

Per a la construcció del graf es va recorrent la llista de diccionaris esmentada anteriorment. En el cas que únicament hi hagi un element en el diccionari, es crea un vèrtex amb el caràcter i la probabilitat que es representa en aquest, i a continuació, s'insereix en el subconjunt corresponent del graf. En cas contrari, es creen tants subgrafs com caràcters sortint representats en el diccionari. Aquest subgrafs, són grafs amb forma d'arbre i representen totes les combinacions possibles de les diferents lletres del subconjunt. Les connexions de l'arbre s'indiquen en els vèrtexs que el formen. Cal destacar, que en aquestes connexions es defineix una puntuació que determinarà grau d'optimització de l'ordenació de les lletres, és a dir, que a la lletra representada sigui succeïda per la lletra definida en el vèrtex de la connexió. A major puntuació la connexió en qüestió serà més òptima. Cadascun dels subgrafs comença per una lletra diferent del diccionari, amb l'objectiu d'obtenir totes les combinacions possibles de les diferents lletres del subconjunt de la paraula. El primer vèrtex de cada subgraf és guardat en l'estructura del graf, mantenint l'ordenació que determina el subconjunt de la paraula que representa. En la Il·lustració 3 es mostra un exemple de com serien els diferents subgrafs creats per a la representació del primer subconjunt de la paraula "weatherproofing". El primer subconjunt estaria format per les tres primeres lletres ("wea").

Com podem anar veient, la complexitat del problema



Il·lustració 2. Exemple dels vèrtexs definits per cada subconjunt de la paraula generosity

recau en determinar l'ordenació de les lletres que estan representades en l'histograma característic. Únicament, la correcta ordenació de les lletres donarà com a resultat una descodificació correcta. Aquest fet implica que en una paraula de 10 lletres, hi ha 3.628.800 maneres diferents d'ordenar les lletres (factorial de 10), no obstant això, el fet de treballar amb histogrames que representen diferents regions, la complexitat de l'ordenació es veu bastant reduïda, existint una possibilitat d'ordenació de 31 maneres diferents. Per contrarestar aquesta complexitat i garantir un major percentatge de solucions correctes, s'ha utilitzat una estratègia basada en la ponderació de les connexions en funció dels digrames que formen. És a dir, a partir d'un llistat dels 75 digrames més comuns a la llengua anglesa, la puntuació de la connexió entre dos vèrtexs està determinada per si el digrama que formen es tracte d'un digrama comú, o no. En cas afirmatiu la puntuació de la connexió es pondera amb 100, en cas contrari, es pondera amb 0.

2.3 Model 2: Arbre complet.

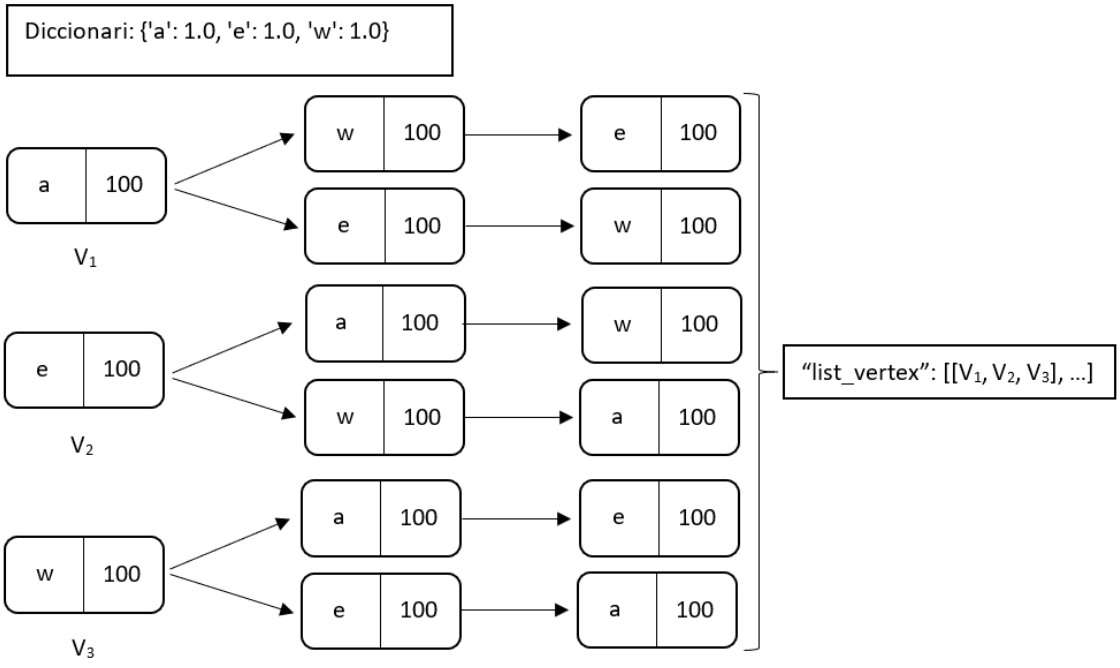
Aquest model neix d'una iteració del model anterior amb l'objectiu de millorar els resultats obtinguts. Les estratègies i l'estructura de dades són bastant similars on la principal diferencia recau en la construcció del graf. S'ha eliminat la diferenciació i creació de petits grafs per cada subconjunt, i en contraposició, es defineixen tants grafs en forma d'arbre com lletres hi hagi en el primer subconjunt de la paraula. El vèrtex inicial de cadascun dels grafs és una lletra de la primera regió de la paraula. A continuació s'explicarà en què consisteixen les modificacions aplicades, tant en l'estructura de dades com en la construcció del graf.

2.3.1 Estructura de Dades

En la segona iteració del model, les classes definides són les mateixes: "Vertex" i "Graph" a tres generals exerceixen una funcionalitat molt similar a la desenvolupada en el model anterior. Pel que fa a la classe "Vertex", mentè els mateixos atributs i els procediments no s'han vist modificats respecte a la primera iteració, a diferència de la classe "Graph", que si s'ha vist modificada, en els aspectes que es mencionen a continuació.

Referent als atributs de la nova classe "Graph" s'ha vist modificada l'estructura utilitzada per gestionar els diferents vèrtexs que componen el graf. A diferència del primer model, el graf únicament guarda una llista dels possibles primers vèrtex, els quals representen el primer subconjunt de la paraula. En cas de la paraula "generosity", exemple mostrat en la Il·lustració 2, en l'estructura d'aquesta iteració únicament es guarden els vèrtexs que representen la lletra "g" i la lletra "e", lletres que formen el primer subconjunt de la paraula. A partir d'aquests vèrtexs i les seves connexions s'obté el camí òptim. Així doncs, tal com s'ha mencionat anteriorment, en aquest exemple podem diferenciar entre dos grafs en forma d'arbre, on el vèrtex inicial d'un dels grafs està representat per la lletra "e" i l'altre per la lletra "g".

En aquesta iteració l'estratègia de cerca del camí òptim s'ha vist bastant modificada. Si en el model anterior arribàvem a estudiar el camí òptim de fins a cinc grafs diferents i la unió d'aquests donava com a resultat la paraula descodificada, en aquest model s'estudia el camí òptim dels grafs representats per les lletres de la primera regió. No obstant això, la complexitat d'aquests grafs és molt més elevada, i l'obtenció del camí òptim ve donada per una exploració completa del graf, sent una estratègia molt costosa en memòria.



Il·lustració 3. Exemple dels sub-grafs resultants del primer subconjunt de la paraula "weatherproofing"

2.3.2 Creació del graf

Com ja s'ha mencionat anteriorment, en aquest segon model la principal diferència recau en la construcció del graf, on a partir dels primers vèrtexs del graf s'estén la resta de vèrtex donant un graf en forma d'arbre. No obstant això, la lectura i representació dels histogrames característics de la paraula a descodificar es manté. En aquesta iteració seguim treballant amb els histogrames del primer, segon i cinquè nivell en funció de la longitud de la paraula. D'igual manera que es segueix utilitzant un diccionari per a cada subconjunt de la paraula on es veu representat les possibles lletres del subconjunt i la seva probabilitat.

Així doncs, a l'hora de construir el graf, novament es recorre la llista de diccionaris de cada divisió de la paraula, i en cas que es tracti del primer subconjunt, els vèrtexs que representen les diferents lletres, seran els primers vèrtexs de l'arbre essent guardats en l'estructura del graf. En aquests primers vèrtexs es creen totes les combinacions possibles de totes les primeres lletres, on cadascuna d'aquestes comença amb una lletra diferent de les possibles del subconjunt. En la següent iteració, novament es creen totes les combinacions possibles i aquestes són annexades com a veïns a cadascun dels vèrtexs finals dels arbres construïts fins al moment. Aquesta estratègia es va repetint per a cada una de les divisions de la paraula. En la Il·lustració 4 es mostra un exemple de com quedaria el graf per a la paraula "lustily".

Novament les connexions dels diferents vèrtexs és ponderada en funció del digrama que formen les lletres representades, utilitzant el diccionari de digrames de la iteració anterior i aplicant la mateixa estratègia de ponderació, 100 punts o 0 en funció de si es tracta d'un digrama característic o no.

La problemàtica principal d'aquesta metodologia és la dimensió que arribar a adoptar el graf, ja que els diferents grafs creats tenen tantes connexions com possibles maneres hi ha d'ordenar els caràcters de la paraula. Per aquest motiu, la seva execució per a paraules superiors als 21 caràcters no ha estat possible com a conseqüència de desbordaments en la memòria en la cerca del camí òptim.

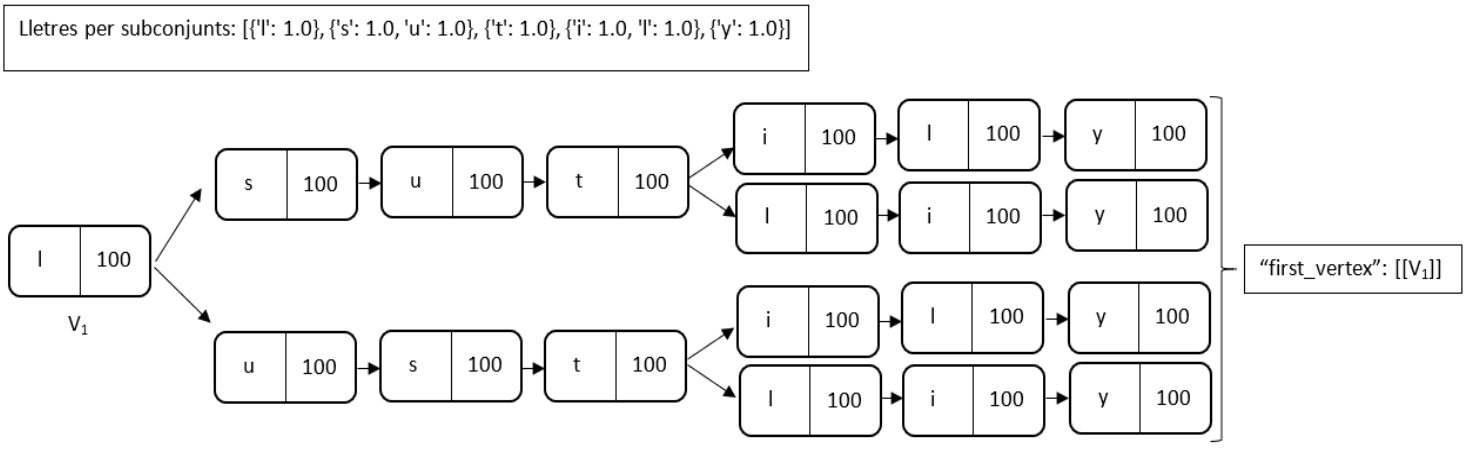
2.4 Model 1 vs. Model 2

Per tal de comprovar i verificar el funcionament dels dos models desenvolupats s'ha efectuat un test basat en la descodificació de 88.173 paraules prèviament codificades en els vectors de característiques descrits anteriorment. La dimensió de les paraules va des d'un caràcter fins a 23 caràcters. El procediment de les proves ha consistit a agrupar el total de les paraules a partir de la seva dimensió per tal d'obtenir els percentatges d'encert per cadascun dels grups. Cal destacar, tal com s'ha mencionat anteriorment, que el segon model no és capaç de descodificar paraules de més de 21 caràcters.

En la Taula 1 es mostren els percentatges d'encert de cadascun dels models. Podem observar com tant pel model 1 com pel model 2 el percentatge d'encert per a paraules de 5 o menys caràcters és del 100%. Aquest resultat és gràcies al fet que treballem amb histogrames de fins a cinc nivells i per tant, en aquest grup de paraules cada caràcter queda representat en un histograma diferent, suprimint la necessitat de discernir entre la posició dels diferents caràcters de la paraula.

Els dos models amb paraules majors de 5 caràcters perden efectivitat, reduintse fins al 53% amb paraules de 6 caràcters, pel que fa al Model 1, i en el cas del Model 2 el percentatge s'ha reduït fins al 74%. El percentatge es va reduint fins a arribar al 0% d'encert en paraules de 14 caràcters, en el cas del primer model, i paraules de 17 caràcters pel que fa al segon model. Podem observar com en el segon model entre les paraules de 7 i 11 lletres és capaç d'encertar entre un 16% i un 19% més vegades que la primera implementació. És en aquests grups de paraules on la millora és més notable. Són resultats satisfactoris, ja que, amb petites modificacions de l'algoritme inicial s'ha aconseguit millorar bastant el rendiment. A més a més, el segon model és capaç de rascar petits percentatges d'encerts en paraules on la primera iteració no obté cap resultat positiu.

Aquests resultats ens confirmen que l'opció de treballar amb un únic graf connectat és més positiu que l'estratègia de crear petits grafs per cadascun dels subconjunts de la paraula. No obstant això, el segon model s'ha d'anar seguint polint, ja que, a hores d'ara està molt poc optimitzat, fins al punt que s'impossibilita el seu funcionament



Il·lustració 4. Graf resultant de la paraula "lustily"

DIMENSIÓ	ACURACY MODEL 1	ACURACY MODEL 2
1	100 %	100 %
2	100 %	100 %
3	100 %	100 %
4	100 %	100 %
5	100 %	100 %
6	53 %	74 %
7	38 %	54 %
8	20 %	36 %
9	11 %	31 %
10	6 %	24 %
11	2 %	12 %
12	2 %	8 %
13	1 %	5 %
14	0 %	3 %
15	0 %	1 %
16	0 %	1 %
17 o +	0 %	0 %

Taula 1. Comparativa d'enecerts entre el Model 1 i el Model 2

per a paraules de moltes lletres.

D'altra banda m'agradaria remarcar com la utilització dels digrames és essencial per a la resolució del problema. Per exemple, en paraules de 10 caràcters hi ha una probabilitat del 50% d'encertar l'orde de les lletres en els diferents subconjunts, ja que cadascun està format per dos caràcters. Aquest fet implica que la probabilitat d'encertar l'ordenació de les lletres de manera completament aleatòria és del 0,03% ($0{,}5^5$). En canvi, gràcies a prioritzar l'ordre en funció de la creació de possibles digrames l'encert ha augment fins al 24% (800 vegades superior) en la segona iteració de l'algoritme.

Si mirem els resultats obtinguts de manera global, tal com es mostren en la Taula 2 on es presenten el nombre total d'encert i el nombre total d'errors, en els dos models implementats fins al moment. Podem observar com el percentatge d'encerts ha augmentat un 14% aproximadament, passant d'un encert del 31,68% a un encert del 45,24% entre el Model 1 i el Model 2.

MODEL	ENCERTS	ERRORS	ACURACY (%)
1	27.937	60.235	31,68
2	39.890	48.282	45,24

Taula 2. Comparativa del total d'encerts i errors entre els dos models

2.5 Model 3: Beam Search.

En aquest segon procés d'implementació s'han modificat algunes de les estratègies utilitzades fins al moment, i s'han introduït nous plantejaments per intentar millorar els resultats obtinguts. Aquests canvis introduïts han afectat en la creació del graf, en la cerca del camí òptim i en el nombre i qualitat dels digrames utilitzats. Les modificacions implementades s'exposen en les següents seccions de l'article.

2.5.1 Optimització dels digrames.

Com s'ha exposat en seccions anteriors, un dels mecanismes que afavoreixen un millor resultat en la descodificació consisteix en el càlcul del cost en funció dels digrames més comuns. En els models anteriors s'ha utilitzat un total de 75 digrames, que constituïen els 75 digrames més comuns de la llengua anglesa, i tots aquests tenien la mateixa puntuació associada, concretament 100 punts. Entenïem que el nombre de digrames era un nombre massa petit i que cadascun d'ells sigues tractat amb la mateixa ponderació podia esbiaixar els resultats que s'estaven obtenint. Per aquest motiu s'ha optat per crear una nova llista de digrames utilitzant el diccionari de 88.173 paraules amb el que es fan les proves de rendiment. S'han obtingut tots els digrames diferents que hi apareixen, formant una llista de 751, a més a més, cadascú d'aquests digrames té associat una puntuació que està determinada pel nombre de vegades que hi apareix en el diccionari. El digrama que més hi apareix es pondera amb 100 punts, i la resta de ponderacions s'han obtingut de manera proporcional a aquest llindar.

Els resultats que s'han obtingut amb aquesta variant han estat bastant satisfactoris, l'increment del percentatge d'encert ha arribat fins a 13 punts per a paraules de 8 lletres. Aportant una millora característica en cadascun dels grups de paraules. En la Taula 3 s'hi presenta una comparativa dels resultats entre el Model 2 implementat amb la llista de 75 digrames, i aplicant la nova estratègia, una llista de 751 digrames.

Novament, podem observar els resultats de manera global, sense diferenciar entre les dimensions de les paraules, tal com es mostra en la Taula 4. Els resultats obtinguts ens indiquen que s'ha pogut incrementar l'encert en

DIMENSIÓ	ACURACY 75 DIGRAMES	ACURACY 751 DIGRAMES
1	100 %	100 %
2	100 %	100 %
3	100 %	100 %
4	100 %	100 %
5	100 %	100 %
6	74 %	79 %
7	54 %	61 %
8	36 %	49 %
9	31 %	40 %
10	24 %	32 %
11	12 %	18 %
12	8 %	9 %
13	5 %	5 %
14	3 %	2 %
15	1 %	1 %
16	1 %	0 %
17	0 %	0 %

Taula 3. Comparativa de resultats amb les dues estratègies de digrames

5.562 paraules, aquest increment representa una millora del 6%, arribant a un percentatge d'encert del 51,59%. Reafirmant novament, que l'estratègia de la utilització de digrames és molt positiva per a la resolució d'aquest problema.

#DIGRAMES	ENCERTS	ERRORS	ACURACY
75	39.890	48.282	45,24 %
751	45.452	42.651	51,59 %

Taula 4. Comparativa dels resultats globals de les dues estratègies de digrames

2.5.2 Optimització del graf

En aquesta iteració del projecte una de les millores s'ha centrat a utilitzar una nova estratègia per a la construcció del graf, aquesta millora ve de la mà d'una nova tècnica per a obtenir el camí òptim d'aquest, basada en l'aplicació de l'algoritme Beam Search que s'exposa en les pròximes seccions de l'article.

L'optimització del graf ha consistit a crear un graf dirigit, on s'elimina la forma d'arbre, i en contraposició, s'implementa una connexió completa entre el vèrtex del mateix nivell de la paraula, o els vèrtexs del mateix nivell estan connectats de manera completa, i també estan connectats a tots els vèrtexs del següent ni-vell. No arriba a ser un graf complet, ja que els vèrtexs de diferents nivells únicament estan connectats d'un nivell inferior cap a un nivell superior. D'aquesta manera mantenim l'ordre dels diferents subconjunts de la paraula que ens proporciona l'histograma. En la Il·lustració 5 és mostra un exemple de com quedaria el graf per la paraula "generosity". Per poder aconseguir aquesta implementació juntament amb la nova estratègia de cerca, el vèrtex en la classe del graf són guardats per nivell, a diferència del model anterior on únicament guardàvem els vèrtexs que representaven el primer subconjunt de la paraula.

Aquesta nova implementació ha donat lloc a la construcció d'un graf molt més reduït i òptim que en el model anterior, juntament amb la implementació del nou algoritme per a la cerca del camí òptim, permet l'execució de l'algoritme per a totes les paraules del subconjunt d'entrenament. Eliminant la impossibilitat de l'execució per a paraules de més de 21 lletres.

L a nova estratègia s'ha pogut implementar mantenint bastant l'estructura dels procediments implementats fins al moment i efectuant petits canvis pel que fa a l'estructura de dades del graf i a la connexió dels diferents vèrtexs d'aquest. En la secció Diagrames de Flux de l'apèndix es presenta el diagrama dels procediments que han donat lloc a la implementació d'aquesta optimització.

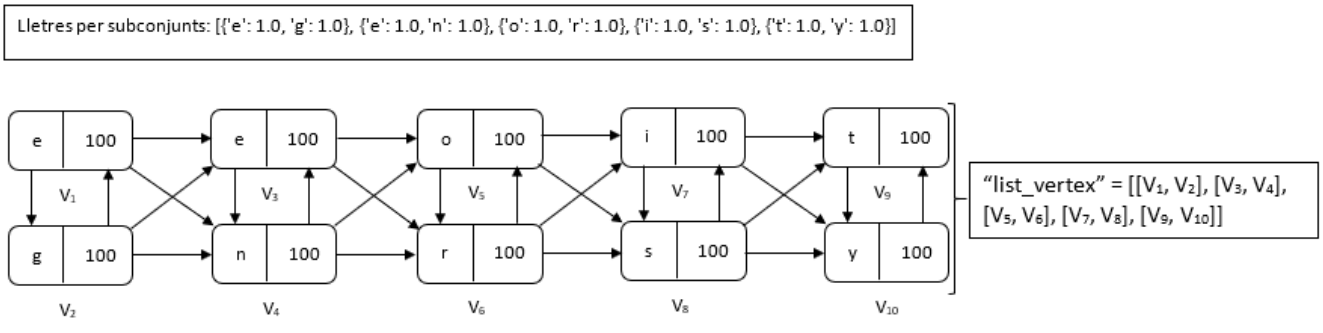
2.5.3 Optimització del camí òptim

Tal com s'ha mencionat en seccions anteriors, aquesta iteració s'ha centrat en l'optimització de l'algoritme per aconseguir una correcta implementació i execució eliminant les limitacions del model anterior per paraules d'una gran llargària. L'optimització també es centra en la cerca del camí òptim, on s'ha implementat un algoritme basat en l'estratègia Beam Search [4]. Aquesta consisteix a eliminar part de les solucions en funció de la seva heurística en cada nivell de l'execució. D'aquesta manera, no mantenim en temps d'execució aquelles solucions que estan donant una baixa probabilitat d'èxit, afavorint molt l'optimització de l'algoritme, pel que fa a la memòria màxima consumida. Fet que impedia la correcta excussió per a paraules de grans dimensions. La poda de les solucions es produeix en el canvi de subconjunt del graf. Per aquest motiu s'ha optat per guardar els vèrtexs mantenint la jerarquia del nivell on estan representats.

D'igual manera que en iteracions anteriors del projecte, les diferents solucions (camins) estan representats per una dupla de dos elements, on en el primer element guardem una llista dels diferents vèrtexs que formen el camí d'aquella solució, ordenats en l'ordre en què s'han visitat. I el segon valor guarda la puntuació acumulada d'aquell camí. Aquests camins es van guardant de manera ordenada, de més puntuació a menys. Un cop s'han actualitzat tots els camins possibles amb els nous vèrtex del nou nivell estudiat, únicament es mantenen les 5 millors solucions, la resta són eliminades, per a la següent iteració de l'algoritme.

Una de les principals característiques i facilitats que ens proporciona l'algoritme és la possibilitat de modificar el llindar de la poda. Permetent la utilització del llindar que millor s'adapti a les necessitats del problema. Per aquest motiu s'han fet diferents execucions de l'algoritme, on s'ha anat modificant el llindar amb l'objectiu de definir el llindar més òptim pel nostre problema. En la secció: 2.5.5 Llindar de poda, es profunditza més en l'explicació i l'estratègia utilitzada per a la elección d'aquest llindar.

Cal remarcar que el principal inconvenient d'aquest algoritme és que pot no donar com a resultat la solució òptima, com a conseqüència de l'eliminació de solucions



Il·lustració 5. Graf resultant de la paraula "generosity"

parcials. No obstant això, el benefici que ens proporciona quant a optimització és molt més satisfactòria que les possibles solucions òptimes que es poden estar eliminant.

Pel que fa a la implementació d'aquesta estratègia s'han modificat els mètodes corresponents del model anterior de la classe graf. En la secció Diagrames de Flux de l'apèndix es presenta el diagrama dels procediments que han donat lloc a la implementació d'aquesta optimització.

2.5.4 Millores de rendiment obtingudes.

S'ha efectuat una comparativa dels costos, tant de memòria com de temps d'execució, entre el segon model i el tercer model amb un llindar de poda de 5. S'han executat en igualtat de condicions sobre el sistema operatiu Windows 10 en la seva última versió a data (09/06/2019), amb un processador Intel Core I7 de 4ª Generació amb quatre nuclis i vuit fils d'execució a una freqüència de 3,5GH, juntament amb una memòria de 16 GB, en dual "channel" a una freqüència de 1600 MH.

Les proves s'han efectuat analitzant els costos de l'execució de l'algoritme per totes les paraules del diccionari, executant-se un total de 88.172 vegades amb paraules de diferents mides. Tal com s'ha mencionat anteriorment l'algoritme implementat en el segon model, era molt poc òptim per a paraules de més de 17 lletres. És per aquest motiu que els dos algoritmes han estat executats diverses vegades, i en cada iteració la dimensió màxima de les paraules ha estat incrementada en una lletra. En la primera execució s'han agafat totes les paraules entre 1 i 18 lletres, arribant a executar-los amb paraules d'una lletra a 21.

Els resultats obtinguts han estat molt satisfactoris, tal com es mostra en la Taula 5, on podem observar una gran diferència entre el comportament de l'algoritme del Model 2 i l'algoritme de model 3. Mentre que l'algoritme del Model 2 es veu molt afectat per l'increment de les dimensions de les paraules, arribant a una diferència de 366 segons i 3,4GB d'utilització de memòria, entre la primera execució (amb paraules de 18 lletres com a màxim) i l'última execució amb paraules de 21 lletres com a màxim. L'algoritme implementat en el tercer model, no es veu gens afectat amb l'increment de les paraules, en totes les execucions el temps ronda els 70 segons i la màxima memòria utilitzada és de 145 MB.

Cal destacar que el percentatge de paraules de més de 17 lletres en el total de diccionari és del 0,07%. Aquest fet remarcar com la implementació de l'algoritme del segon model es veu molt perjudicada amb paraules de grans dimensions.

2.5.5 Llindar de poda

Un paràmetre que hem d'ajustar per tal d'obtenir la millor relació rendiment–costos de l'algoritme que implementa Beam Search es tracta del llindar de la poda que s'efectuarà en cada nivell d'excussió de l'algoritme. Recordem, que aquest llindar indica quantes solucions parcials mantenim com a màxim al llarg de l'execució, i en conseqüència, la resta de solucions, que en aquell moment tenen una heurística pitjor, són eliminades. Així doncs, la utilització d'un llindar massa baix farà que en molts casos eliminem la solució correcta, impossibilitant una descodificació exitosa de la paraula. D'altra banda, un llindar massa elevat, si bé és cert que el percentatge de solucions correctes millorarà, el seu rendiment quant a temps d'execució i la memòria màxima necessària per a la seva execució incrementarà.

Per tal d'obtenir aquest llindar òptim, s'han efectuat múltiples execucions de l'algoritme, obtenint els percentatges d'encerts i la mitjana dels costos de cadascuna de les execucions, tot variant el llindar entre 1 i 10. Els resultats que s'han obtingut es mostren en l'Apèndix. Com ja va sent habitual, totes les execucions s'han efectuat amb el total de les paraules del diccionari.

Podem destacar com a partir d'un llindar de 5 el percentatge d'encert quasi no varia, en canvi, els costos de l'algoritme segueixen creixent amb la mateixa proporció. Aquest fet, juntament amb què el seu temps d'execució és aproximadament de 66 segons, temps no gaire elevat tenint en compte que estem descodificant un total de 88.172 paraules, que implica una descodificació de 1.335,93 paraules cada segon (1,33 paraules descodificades cada mil·lisegon), podem afirmar que per a la solució d'aquest problema, la millor opció és aplicar l'algoritme amb un llindar de 5.

D'altra banda m'agradaria remarcar els fets que s'han exposat anteriorment, ja que, a la taula de resultats podem observar com llindars molt baixos, 1 o 2, perjudiquen els resultats de la descodificació, obtenint un percentatge d'encert del 46,41 i 49,72 respectiva-ment. A més a més aquesta pèrdua no es veu reconfortada per la disminució dels costos, ja que únicament estem guanyant 6 segons en el temps d'execució. Un altre fet característic és que la memòria màxima necessària no es veu influenciada per aquest llindar, on es manté al voltant dels 145 MB per cadascuna de les execucions.

2.5.6 Els cinc millors resultats.

Gràcies a la implementació d'una estratègia de cerca amb l'aplicació de l'Algoritme Beam Search ha permès obtenir fins a 5 possibles solucions de la paraula a descodificar. Fins al moment únicament en cada descodificació s'obte-

	MODEL 2		MODEL 3 PODA 5	
MAX DIMENSSIO	TEMPS (s)	MEMORIA (MB)	TEMPS (s)	MEMORIA (MB)
18	132,19	330	69,56	145,4
19	206,23	899	71,92	145,3
20	309,62	3.140	68,87	145,5
21	498,47	3.824	69,73	145,4

Taula 5. Comparativa de rendiment entre el Model 2 i el Model 3 amb Poda 5

nia una possible paraula, però amb aquesta nova estratègia, podem arribar a obtenir fins a 5 paraules, les cinc solucions més òptimes de la descodificació.

Aquest fet ha implicat que el percentatge d'encert, si ens centrem en la solució més òptima que ens proporciona l'algoritme, és del 50,54%. En canvi si anem optant per les diferents solucions que obtenim, aquest percentatge va augmentant. En el cas d'obtenir les dues descodificacions més òptimes, arribem a un percentatge d'encert del 68%; si ens fixem en les tres millors solucions, l'encert s'augmenta fins al 75%. Arribant a un encert del 81% obtingut amb les 5 solucions més òptimes que ens proporciona l'algoritme, sent capaços d'incrementar l'encert obtingut per la solució més òptima en aproximadament 27 punts. Aquest resultats s'han obtingut, com ja va sent habitual, executant l'algoritme per cadascuna de les paraules del diccionari. En la Taula 6 és presenten els resultats obtinguts.

SOLUCIÓ	# ENCERTS	# ERRORS	ACURACY
1ra	44570	43602	54,54%
2na	12018	76154	13,63%
3na	6287	81885	7,13%
4rta	3024	85148	3,42%
5éa	2019	86153	2,28%

Taula 6. Percentatge d'encert de les cinc solucions més òptimes, per l'execució del diccionari.

2.5.7 Implementació de l'algoritme en un data set d'imatges

Fins al moment tots els resultats que s'han presentat, s'han obtingut d'executar l'algoritme d'un diccionari de 88.173 paraules literals, i per tant, estàvem treballant amb histogrames binaris evitant el soroll en la lectura de les paraules. No obstant això, l'objectiu del projecte és poder reconèixer i descodificar les paraules obtingudes en imatges mitjançant d'un sistema OCR.

Com a conseqüència del soroll introduït en els histogrames, els resultats que s'han obtingut en el canvi de data set, passant d'histogrames binaris a histogrames probabilístics, s'han vist perjudicats. Fins al moment, com sabíem exactament quines lletres formaven la paraula, la implementació del problema a solucionar se centrava a discernir l'orde de les diferents lletres dins de la mateixa paraula. En aquest punt, la problemàtica és doble, ja que no únicament ens centrem en l'orde de les lletres sinó, que a més a més, en discernir i saber quines són les lletres que formem la paraula.

La solució que s'ha optat per a la resolució d'aquesta nova problemàtica que es planteja, és la de definir un llindar, i a partir d'aquest, considerar si una determinada lletra està representada, o no, en la paraula a descodificar. Així doncs, si un valor de l'histograma superà el llindar marcat, es considera que la lletra que està representant és una lletra de la paraula. Cal destacar, que la definició del llindar està molt lligada al conjunt de dades que s'ha utilitzat per a les proves. Per tant, per unes altres dades, el llindar òptim s'hauria de tornar a calcular.

Per tal de garantir la millor solució òptima per a la resolució del problema, la definició del llindar a utilitzar s'ha obtingut efectuant diferents execucions, variant aquest i comprovant el percentatge d'encert obtingut en cada cas. Tal com es mostra en Taula 7 podem determinar que el llindar que ens ha proporcionat un millor rendiment és del 0,4, obtenint un percentatge d'encert d'aproximadament el 61%, on s'ha aplicat el tercer model implementat, aplicant la llista de digrames ponderats.

Un cop s'han definit les lletres que hi estan representades en l'histograma característic, la resolució del problema ve donada d'igual manera que l'exposada anteriorment en els models.

El percentatge d'encert mostrat en la Taula 7, s'han obtingut únicamente estudiant la solució més óptima que ens proporciona l'algoritme. No obstant això, també s'ha comprovat el rendiment del model aplicant un estudi del les millors solucions amb un llindar de poda 5. En aquest cas podem observar que, tal com es mostra en la Taula 8, novament el percentatge d'ancert augmenta si accaptem les 5 millors solucions que ens proporciona l'algoritme. En aquest cas, augmenta en un 8% el percentatge d'encert amb els dos millors resultats i fins a un 11% amb els tres millors resultats. Podent arribar a un encert del 73% amb els cinc resultats més óptims que ens proporciona l'algoritme.

LLINDAR	# ENCERTS	# ERRORS	ACURACY
0,1	222	324	40,66 %
0,2	286	260	52,38 %
0,3	318	228	58,24 %
0,4	333	213	61,17 %
0,5	330	216	60,44 %
0,6	306	240	56,04 %
0,7	290	256	53,11 %
0,8	260	286	47,62 %
0,9	199	347	36,45 %

Taula 7. Percentatge d'encert en funció del llindar

SOLUCIÓ	# ENCERTS	# ERRORS	ACURACY
1ra	334	212	61,17 %
2na	44	502	8,05 %
3na	18	528	3,29 %
4rta	1	545	0,18 %
5éa	5	541	0,91 %

Taula 8. Percentatge d'encert de les cinc solucions més òptimes

3 CONCLUSIÓ

Al llarg del projecte s'han anat implementant diferents models per tal d'assolir els objectius establerts. Cadascun dels models implementats s'urgeixen d'una iteració del model anterior, amb l'objectiu d'anar millorant els resultats i el rendiment model a model. Els models desenvolupats s'han basat en una estratègia de creació d'un graf i una posterior cerca de camins òptims, on les estratègies de construcció i d'obtenció del millor camí s'han anat modificant, per anar millorant els resultats i l'optimització dels algoritmes.

Cal destacar, que aquest projecte neix com una iteració del projecte "Word Spotting and Recognition with Embedded Attributes", on mitjançant tècniques de visió per computador, un etiquetatge molt ajustat i amb l'ajuda d'un diccionari, aconsegueixen interpretar paraules a partir d'imatges naturals. No obstant això, el model únicament era capaç d'interpretar aquelles paraules que hi apareixien en el diccionari. Així doncs, amb la realització d'aquest projecte, s'ha aconseguit prescindir del diccionari per a la interpretació. Aconseguint un percentatge d'encert bastant satisfactori, que pot arribar fins al 70% amb la combinació de les solucions més òptimes que ens proporciona el model implementat.

De cara a millores futures, la implementació del model seria fàcilment escalable si a l'hora de construir el graf, tinguéssim la informació de la dimensió de la paraula que estem descodificant. Aquest fet ajudaria a l'hora d'acceptar, o no, les lletres que es veuen representades en l'histograma característic, aconseguint ser molt més fidelignes a la paraula a descodificar incrementant els percentatges d'encerts obtinguts.

BIBLIOGRAFIA

[1] J. Almazán, A. Gordo, A. Fornés i E. Valveny (2014) "Word Spotting and Recognition with Embeddaded Attributes" *IEEE TRans. Pattern Anal. Mach. Itell., vol 36*

[2] M. Gamon (2006) "Graph-Based Text Representation for Nvelty Detection" Workshop on TextGraphs, at HLT_NAACL, pp 17-24.

[3] "Paython" URL: http://www.python.org/

[4] C. Tillmann i H. Ney (2003) "Word Reordering and a Dymanic Programming Beam Search Algorithm for Statistical Machine Translation" *Computacional Linguistics., col 29, pp. 97-133*

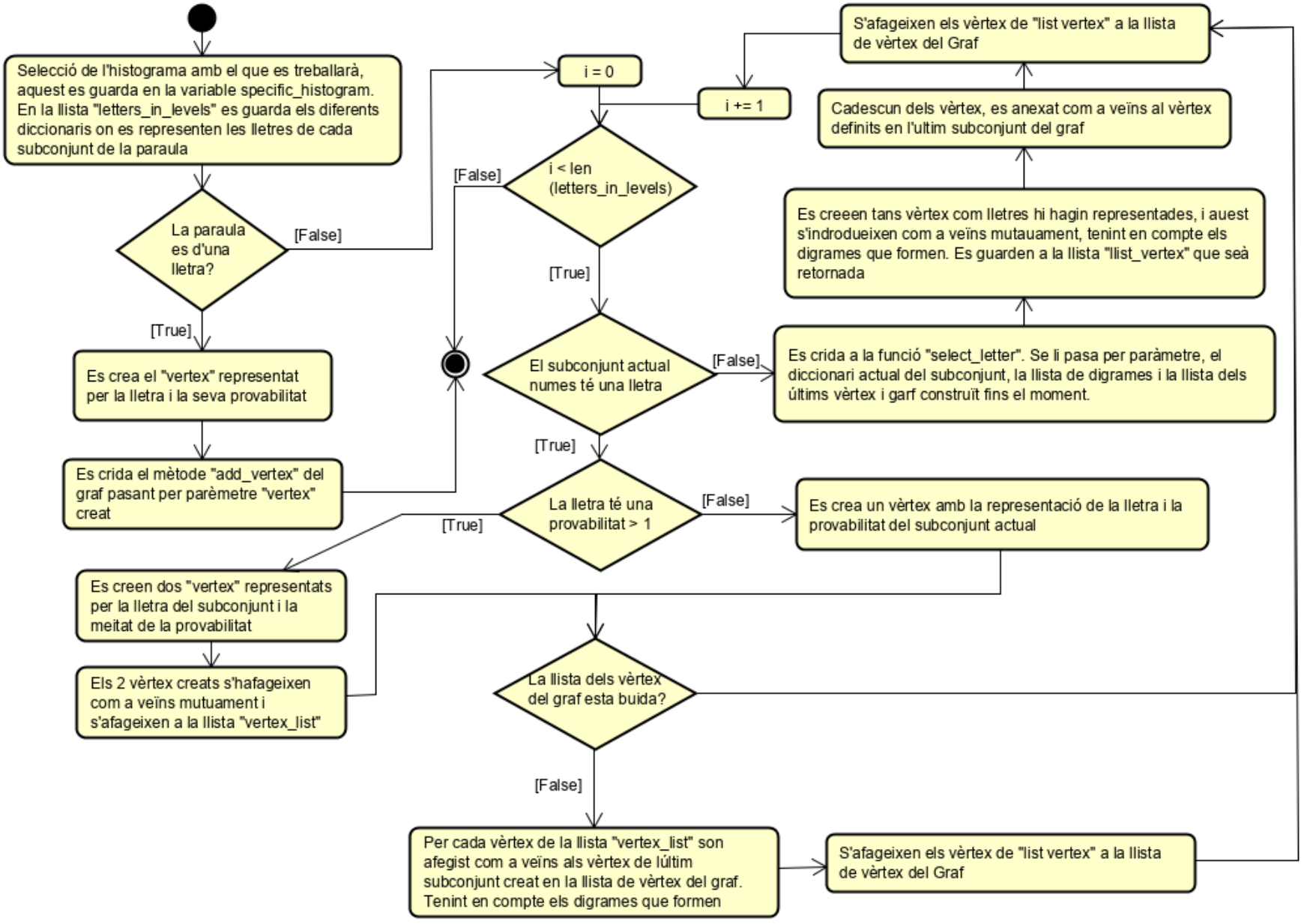
APÈNDIX

A1. RESULTATS DE LES EXECUCCIONS PER A L'ELECCIÓ DEL LLINDAR DE PODA

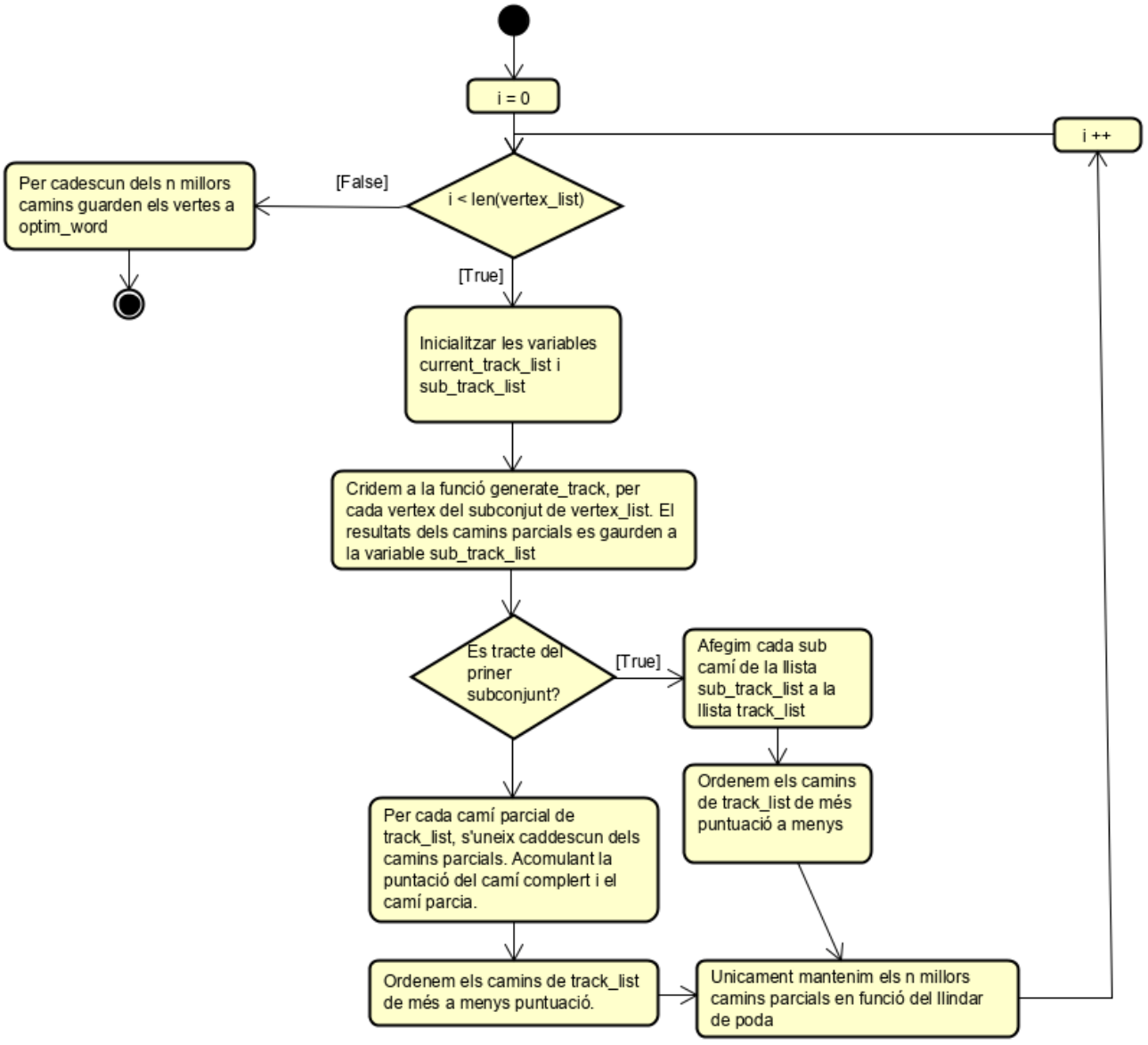
LLINDAR	ENCERTS	ERROR	ACURCY (%)	TEMPS (s)	MEMORIA (MB)
1	40922	47250	46,41	60,69	145
2	43841	44331	49,72	62,53	145,2
3	43803	43803	50,32	65,29	145,5
4	44449	43723	50,41	68,56	145,7
5	44570	43602	50,55	66,24	145,9
6	44576	43596	50,56	69,29	145,9
7	44586	43586	50,57	73,7	145,4
8	44593	43579	50,58	75,51	145,4
9	44604	43568	50,59	76,15	145,4
10	44603	43569	50,59	78,67	145,7

Taula 1. Resultats de les execucions de l'algoritme variant el llindar de poda de 1 a 10

A2. DIAGRAMES DE FLUX



Il·lustració 1. Diagrama de flux de la generació del graf del Model 3



Il·lustració 2. Diagrama de flux de l'algoritme de cerca del camí òptim pel model 3