NASARI

Prima risorsa vettoriale che studiamo. Pensiamo a nasari come ad un insiemi di vettori che invece di essere dei descrittori di termini (word embeddings) sono dei descrittori di sensi (sense embeddings). Gli elementi che descrivono un vettore non sono di tipo numerico ma concettuali.

Come è stato costruito NASARI?:

* E’ stato costruito a partire da wikipedia. Hanno preso i termini ai quali sono stati mappati i synset di wordnet, babelsynset, sono stati presi l’insieme di testi all’interno della pagina e da li sono stati estratti i vettori. L’elemento interessante è che NASARI è nata con 2 tipi di vettori

Due tipi di vettori:

* **vettori lessicali:** rappresenta un certo senso attraverso un elenco di item lessicali. Play theater è un elenco di termini associati a play inteso nell’accezione di teatro. Con play activity fornisco un elenco di termini associati a play nell’accezione di attività.
* **vettori unified:** I sensi sono definiti attraverso una lista di altri sensi.

In entrambe le varianti gli elementi che sono più vicino alla testa del vettore ovvero al senso che il vettore descrive sono progressivamente più importanti avvicinandosi alla testa e meno importanti verso la coda nella definizione del senso. Se parliamo di play drama, il sysnet di play drama sarà tanto più in alto quanto più rilevante nella definizione di play theater.

Algoritmo sviluppato per la creazione di NASARI:

Dopo aver raccolto tutti i termini rilevanti nella pagina di wikipedia, l’algoritmo calcola della misure come la lexical specificity, che sono i pesi della risorsa, e la caratteristica finale è che tutti questi elementi collocano i due tipi di vettori all’interno di uno spazio comune. I vettori così costituiti continuano ad essere confrontabili anche se non sono numerici.

Avendo dei vettori non numerici, capire la similarità tra sensi, in ottica di disambiguazione diventa difficile. La metrica usata è il weighet overlap. Sommatoria su tutte le feature condivise dai due sensi in questione e dal rango della feature condivisa nel primo vettore + rango della feature nel secondo. Il rango è la posizione della feature nel vettore (da 0 a N feature), elevato a -1. Poi per ogni senso nei due vettori si calcola il peso, infine si prendono il weighted overlap massimo.

Vettori lessicali: il formato è:

* babel synset id
* opzionalmente il titolo della pagina di wikipedia
* coppie di elementi (lemma, peso)

Vettori unified:

* Babel synset id
* Opzionalmente titolo della pagina di wikipedia
* Invece di avere delle coppie (lemma, peso), abbiamo coppie (babelsynsetid, peso)

Vettori embedded:

* Costituiti da un numero fisso di dimesione, 300 elementi

Automatic summarization: (riassunto automatico)

Esempi di riassunti sono: abstract di un articolo scientifico, bignani... Diverse finalità, può essere indicativo, per selezionare elementi. Finalità informativa: per semplificazione. Finalità critiche, consente di mettere in rilievo alcuni elementi a partire da una visione.

Tipi di riassunti automatici:

* **Estrattivo**: il riassunti viene creato prendendo porzioni del testo originale, identificate come le più importanti
* **Astrattivo**: si fa un modello di quello che ci è scritto e poi si va a generare un nuovo documento

Tipi di uso del riassunto sono:

* Focalizzato su una classe di utenti
* (domanda possibile, quanta informazione è necessaria per fare il riassunto? E quanta informazione ha a disposizione il lettore)
* Il destinatario, la comunità di coloro che ricevono il riassunto

2 approcci:

* **Approccio shallow,** statistico, si limitato alla produzione di riassunti estrattivi. Utilizzati per estrarre gli elementi rilevanti nel documento originale, sulla base di alcuni criteri di rilevanza (frasi, pezzi di paragrafi) non alterando la posizione di tali parti nel riassunto rispetto alla loro posizione nel documento originale. Approccio basato su conteggi. Si opera a livello lessicale (sintattica), si prendono dei termini e si cerca di cerca di impilare sequenze di termini avendo criteri più fini come n-grammi. Richiedono delle qualità di testo consistenti ma oggi le capacità di storage sono alte. Il vantaggio è che sebbene non sono basati su approcci che individuano la semantica sono piuttosto robusti. Funzionano ma la qualità è inferiore.
* **Approccio deeper**, puntano a caratterizzare in un qualche formato logico la semantica della frase, si avvalgono di strumenti di generazione del testo. Sono approcci che sono possibili in contesti molto specifici e circorscitti (bollettino del meteo...) Costuisce una rappresentazione interna, logica, dei significati delle frasi. Se funziona produce dei riassunti di qualità più alta. Viene rigenerato un testo come farebbe un essere umano. Ma molto lento. Non adatto a tutti i contesti.

Due altre divisioni:

* Sorgente singola: un documento molto lungo che si vuole accorciare rispetto ad una percentuale
* Sorgente multipla: dati più documenti dello stesso argomento si tratta di ricostruire il significato del documento considerando più documenti.

I parametri del riassunto automatico:

* **Tasso di compressione**: rapporto tra lunghezza del riassunto sulla lunghezza dell’originale
* **Audience**: destinazione del riassunto, (specifico o generico)
* **Relazione con la sorgente**: (estrattivo o astrattivo)
* **Funzione**: se indicativa o informativa o critica
* **Coerenza**: dato un testo, se cancello delle frasi meno rilevanti potrei aver tolto frasi la cui mancanza mi crea dei buchi nel ragionamento oppure semplicemente lascia incomplete delle anafore. Ad esempio cancello una frase in cui occorre un nome proprio e lascio quella a seguire dove ci si riferisce al nome con : “egli” “lui”... ecc

Come individuare gli elementi rilevanti nelle frasi o le frasi più rilevanti del testo in input?

* **In base alla posizione nel testo.** Ci sono dei luoghi in cui si addensano i termini più rilevanti o meno. Si prendono tendenzialmente i primi pezzi del documento e quelli finali. Si assume che nei testi frequentemente, in molti testi, l’informazione rilevante ricorre ad esempio in posti tipici. Ad esempio in un articolo scientifico, leggo l’abstract e salto alle conclusioni. Non funziona per i romanzi. Però, l’85% di argomenti di frasi importanti stanno nei posti iniziali del documento e il 7% nella fine.
* **In base al titolo:** Il titolo può essere indicativo e sintetico del contenuto del documento. Funziona negli articoli di giornale. Si prende il titolo, si prendono le parole più importanti e usarle come keyword per studiare le frasi che hanno quelle keyword. Sulla base della coesione di un paragrafo con gli elementi del titolo posso dire se sono più o meno rilevanti per il riassunto. Non funziona quando i titoli non sono parole. Si prendono le parole chiavi dal titolo. In base a queste parole è possibile comprimere il testo del documento
* **In base al genere**: (optimum policy) le frasi rilevanti possono occorrere in posizioni in relazione col genere del documento. Spezzetto il testo in unità poi calcolo l’entropia dei termini in ogni unità e prendere quelle dove c’è più informazione. Overlap tra frasi e termini estratti dal titolo, poi si fa un ordinamento parziale delle frasi e scegliere quelle che hanno il punteggio più alto.
* **Cue phrases. In base ad alcune espressioni**: ad esempio, “come detto in precedenza”, “come abbiamo dimostrato..” Ci permettono di individuare porzioni significative. Si può fare sia con elementi che contribuiscono alla rilevanza sia con locuzioni che contribuiscono alla irrilevanza. Quindi applicare meccanismi di massimizzazione della rilevanza o minimizzazione della irrilevanza
* **In base alla coesione:** La coesione è, dati un insieme di paragrafi da p1 e p9 possiamo immaginare di vedere quanto sono sovrapposti e coerenti i significati degli uni con gli atri. Usiamo una topologia connessa. Si possono fare con delle partizioni, usando frasi più rilevanti... (non stiamo dicendo nulla in specifico.) In base alle coreferenze, co-occorrenze, similarità lessicali (concept similarity). Ad esempio una strategia baseline è studiare le co-occorrenze di insiemi di termini fra paragrafi, TF-IDF ecc... Si possono individuare dei punteggi di affinità fra paragrafi quindi prendere quelli con punteggio più alto.

Esercizio di laboratorio: **automatic summarization:**

* Dal punto di vista della selezione del contenuto possiamo fare delle semplificazioni e muoverci con la granularità delle frasi,
* dal punto di vista dell’ordinamento dell’informazione, non ci poniamo il problema. Se non alteriamo la sequenza degli ordinamenti delle frasi nel documento originale non rischiamo di fare operazioni di riordinamento
* Non approfondiamo la realizzazione delle frasi. Scegliamo solo se una frase è rilevante oppure no. Non modifichiamo le frasi.

**Algoritmo non supervisionato più semplice**: selezionare delle frasi che hanno dei termini che sono salienti ed informativi. L’importanza la calcoliamo usando uno strumento semplice che chiamiamo: **“topic signature”.** Sono un insieme di termini significativi che provengono dal titolo. Tutto quello che ha un punteggio di importanza maggiore di una certa soglia lo consideriamo nel riassunto. L’importanza può essere calcolata per mezzo di word frequency. Ma può creare dei problemi. Oppure prendiamo tutto quello che nel titolo è significativamente legato ai paragrafi del testo da riassumere. Oppure i vettori lessicali di NASARI. Oppure la lexical specificity, invece di cercare i termini più frequenti cerco quelli più discriminativi, che meno occorrono più volte e quindi possono aiutare a riassumere.

Esempio di un tipico algoritmo: Eseguiamo 3 tipi di compressione: 10%, 20%, 30% (10% significa che il testo riassunto resta 90% grande) Diverse operazioni:

1. **Individuazione dell’argomento:** L’algoritmo parte dal titolo, si prendono gli elementi rilevanti dal titolo. Il titolo può essere inteso come una lista di vettori NASARI, della forma: termine1, score1... Quindi vettori lessicali
2. **Ci creiamo un contesto**, raccogliamo i vettori dei termini che abbiamo estratto. Nel caso in cui il titolo è corto e c’è poca informazione,
3. **Pesare i paragrafi** sulla base dei termini più importanti in relazione al titolo, usando la metrica: weighted Overlap. Prendiamo 2 vettori, consideriamo le feature in comune e calcoliamo il rango. I paragrafi vengono pesati e tutto ciò che sta al di sopra delle ipotesi di compressione viene mantenuta nella versione riassunta.

Risorse:

* 2 Distribuzioni di NASARI
  + Dd-nasari.txt, ridotto i vettori a 10 features. 600mb
  + Dd-small-nasari.txt, 15 features. 2mb Creato sulle 60000 parole più importanti della lingua inglese.
* 5 documenti
  + Compressiono 10, 20, 30%

Valutazione:

* 2 metriche: BLEU – ROUGE
  + BLEU: considera la precision: tra i risultati che il mio sistema fornisce guardo i rilevanti. Costruiamo un riassunto di riferimento come lista dei termini rilevanti dei documenti(originale e riassunto)
  + ROUGE considera la recall: tra tutti i risultati quanti sono riuscito a rilevarne. Evitiamo gli n-grammi. Overlapp di elementi singoli. Valutiamo approssimativamente la recall.
* NASARI non usa i verbi. I vettori non rappresentano i verbi.
* 3 versioni di vettori