Coverage

Vediamo adesso una applicazione del mondo reale di quanto abbiamo visto finora. Questo strumento, sviluppato presso il Dipartimento di Informatica dell'Università di Torino viene detto **Cover** perché l'algoritmo su cui si basa si chiama COVERAGE (i.e., *Common-sense Vectorial Representation Automatic GEnerator*).

Lo scopo dell'algoritmo è di integrare alcuni strumenti visti finora (e.g., come BabelNet, ConceptNet) per generare in modo automatico una rappresentazione vettoriale del senso comune (i.e., common sense) per un dato concetto in input.

Riprendiamo un attimo gli elementi più importanti visti finora per comprendere appieno come funzioni Cover.

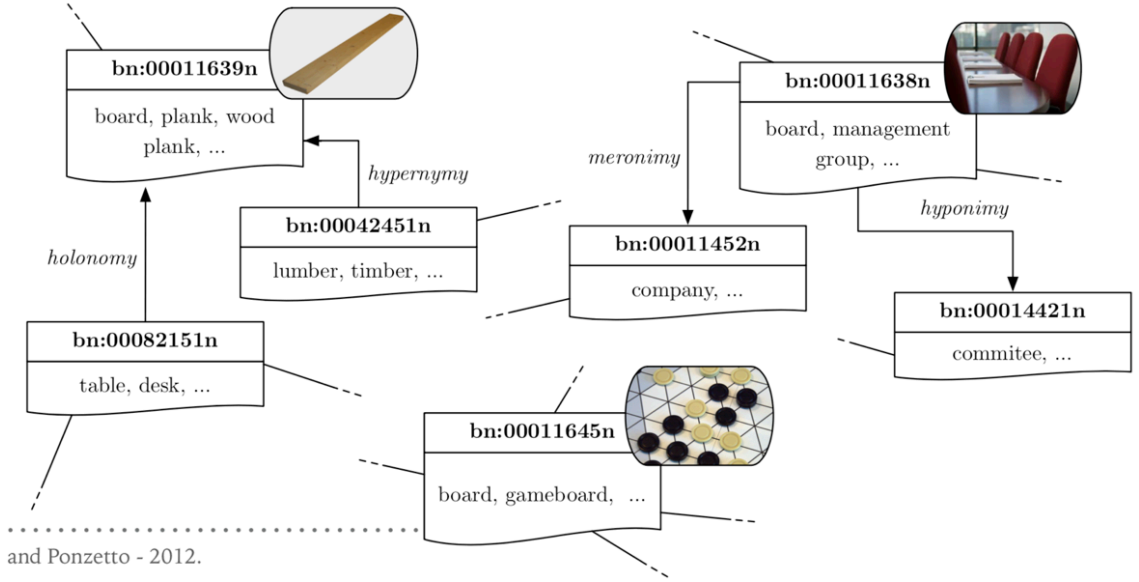
Nota:
Il progetto è stato fatto da: D. Colla, A. Lieto, E. Mensa, D. Radicioni

BabelNet

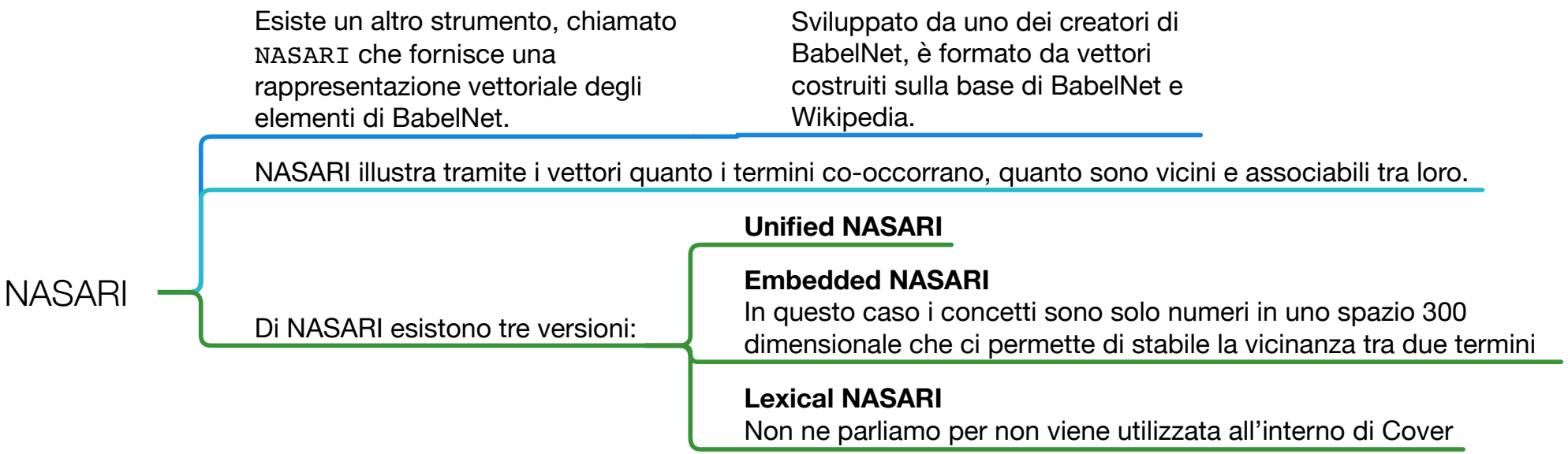
BabelNet è una risorsa vasta, multilingua e automaticamente costruita. È una rete semantica in cui i nodi sono i synset, ovvero insiemi di sensi.

Contiene 14 milioni di synsets

Contiene 746 milioni di sensi di parole (distribuiti tra più di 271 lingue che contiene)



Es. differenti synset che contengono il termine *board*


NASARI
Esiste un altro strumento, chiamato NASARI che fornisce una rappresentazione vettoriale degli elementi di BabelNet.
Sviluppato da uno dei creatori di BabelNet, è formato da vettori costruiti sulla base di BabelNet e Wikipedia.
NASARI illustra tramite i vettori quanto i termini co-occorrano, quanto sono vicini e associabili tra loro.
Di NASARI esistono tre versioni:
Unified NASARI
Embedded NASARI
In questo caso i concetti sono solo numeri in uno spazio 300 dimensionale che ci permette di stabile la vicinanza tra due termini
Lexical NASARI
Non ne parliamo per non viene utilizzata all'interno di Cover

Unified NASARI

Ogni vettore possiede una **testa** che indica quale concetto stiamo analizzando tramite un BabelNet synset id (e.g., `bn:00011639n`) e WordNet synset id (e.g., `wn:15101854n`) insieme al titolo di Wikipedia (e.g., `Plank (wood)`).

Nella **coda** del vettore invece abbiamo una lista di BabelNet synset id correlati che aiutano a rappresentare il concetto. Per ogni elemento presente nella coda associamo un valore (e.g., `[343.35]`) che indica quanto l'elemento sia correlato al concetto contenuto nella testa

```
bn:00011639n    {board, plank, Plank (wood), ...}
wn:15101854n
Plank (wood)

bn:00052293n    [343.35]    {Timbered, 2x4 wood, ...}
bn:00011639n    [289.82]    {board, plank, ...}
bn:00081492n    [249.42]    {wood, sapwood, ...}
bn:00013077n    [201.57]    {bridge, span, ...}
bn:00074531n    [126.16]    {Strake, Wale, ...}
bn:00077259n    [112.47]    {timber}
bn:00008691n    [104.7]     {barrel, cask, ...}
```

TESTA

CODA

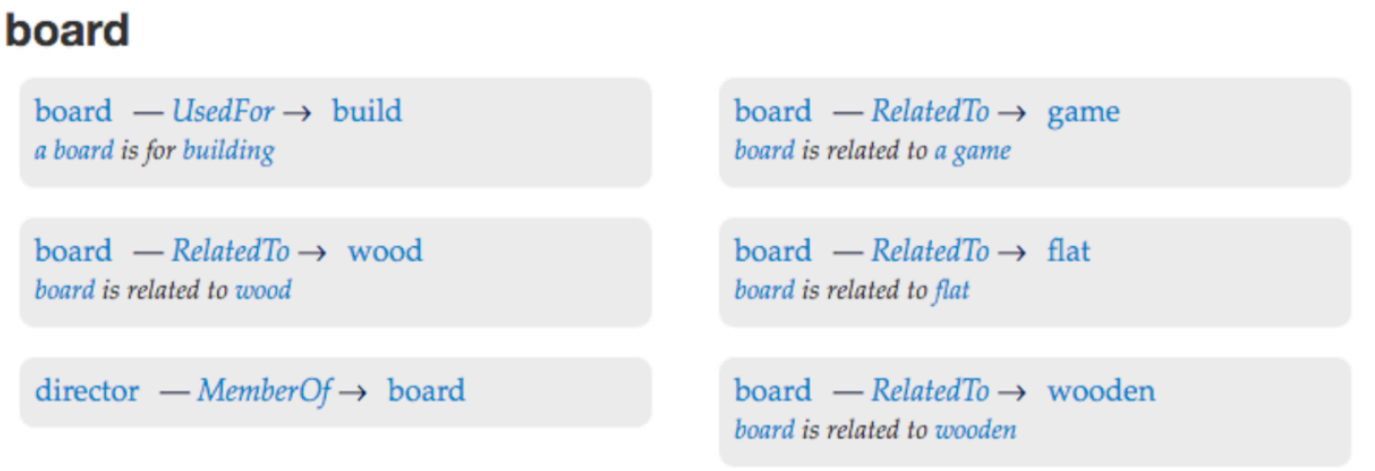
Es. Unified vector per il concetto *board*

ConceptNet

L'ultima grande risorsa che ci interessa in questo momento è ConceptNet

È una rete basata sulla conoscenza di common-sense

Contiene più di 10milioni di relazioni che collegano circa 3milioni di concetti



Pagina per il concetto *board*

Cover

Come possiamo mettere insieme le cose per ottenere una rappresentazione del senso comune? L'idea è la seguente:

1. Cover riceve in input un concetto

Nota:
Da ora in avanti quanto parleremo di concetto intenderemo i BabelNet synset-id

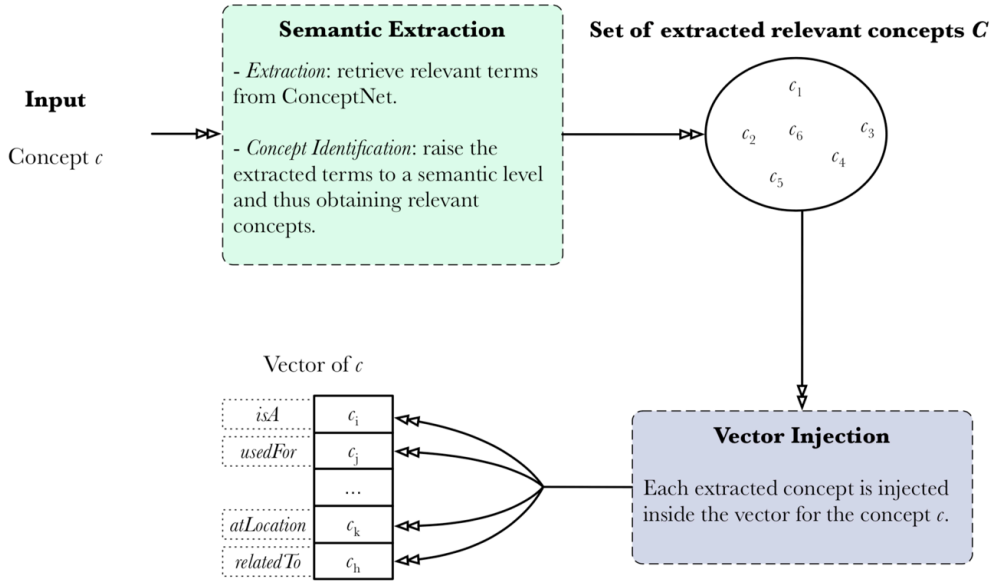
2. Dopo abbiamo una fase di estrazione semantica in cui vengono eseguite due operazioni:

2.1 Una *prima* di estrazioni in cui selezioniamo i *termini rilevanti* da ConceptNet

2.2 Dopo abbiamo una *seconda* fase di identificazione dei concetti dove andiamo ad elevare i termini estratti ad un livello semantico (i.e., *otteniamo la semantica dei termini estratti*) ottenendo cosi i *concetti rilevanti*

3. In seguito andiamo a scrivere l'insieme dei concetti estratti all'interno di un vettore c che conterrà una serie di campi:

relatedTo, isA, atLocation, usedFor, madeOf, ...



relatedTo	*tool, food, utensil, cutlery, eating*
isA	*tool, cutlery, utensil*
atLocation	*table, desk, plate*
usedFor	*eating*
madeOf	*metal*

Es. vettore *fork*

Procedimento descritto

L'aspetto più importante è che l'informazione contenuta all'interno del vettore risultato è concisa, rilevante e inerente il concetto.

Questa esigenza deriva dal fatto che il senso comune è prezioso perché non è enciclopedico e fornisce per l'appunto poca informazioni ma molto utile e rilevante

Esempio
passo
1 - 2.1

Consideriamo la costruzione del vettore per il concetto di **fork** (forchetta).

La prima cosa da fare è prendere il vettore NASARI del termine e procedere alla fase di estrazione con cui cerchiamo in ConceptNet i vari modi di riferirsi alla forchetta secondo i synset di BabelNet.

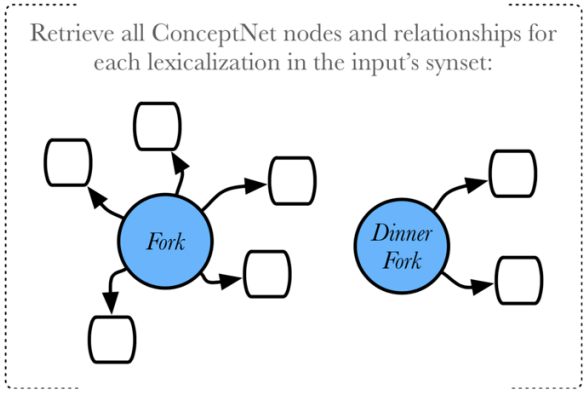
Es. fork in BabelNet si dice anche pickle fork o dinner fork. Ognuno di questi termini deve essere cercato in ConceptNet

Es. L'input del sistema è il concetto *c* `bn:00035902n` che corrisponde al BabelSynsetId di *fork* inteso come "a utensil used for eating or serving food"

bn:00024649n {tableware, …}
bn:00049322n {knife, …}
bn:00073547n {spoon, …}

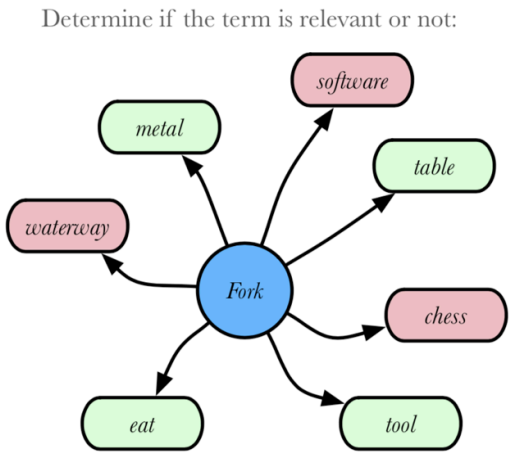
…

Es. parte del vettore NASARI per fork



Esempio
passo
2.1

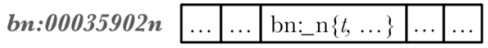
Grazie a questa prima operazione recuperiamo anche l'insieme di termini collegati a fork. Alcuni di questi termini saranno interessanti per noi (e.g., *table*) ma altri invece no (e.g., *il version control system dell'informatica*).



software, waterway e chess non sono rilevanti

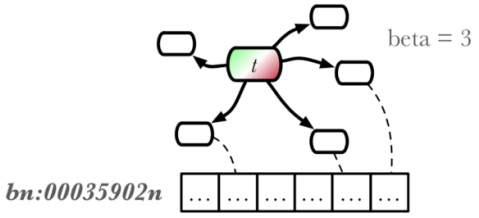
Dunque procediamo con la fase di identificazione dei concetti rilevanti.

Questa fase viene eseguita considerando ogni nodo restituito e terminando se il termine contenuto nel nodo è rilevante o meno



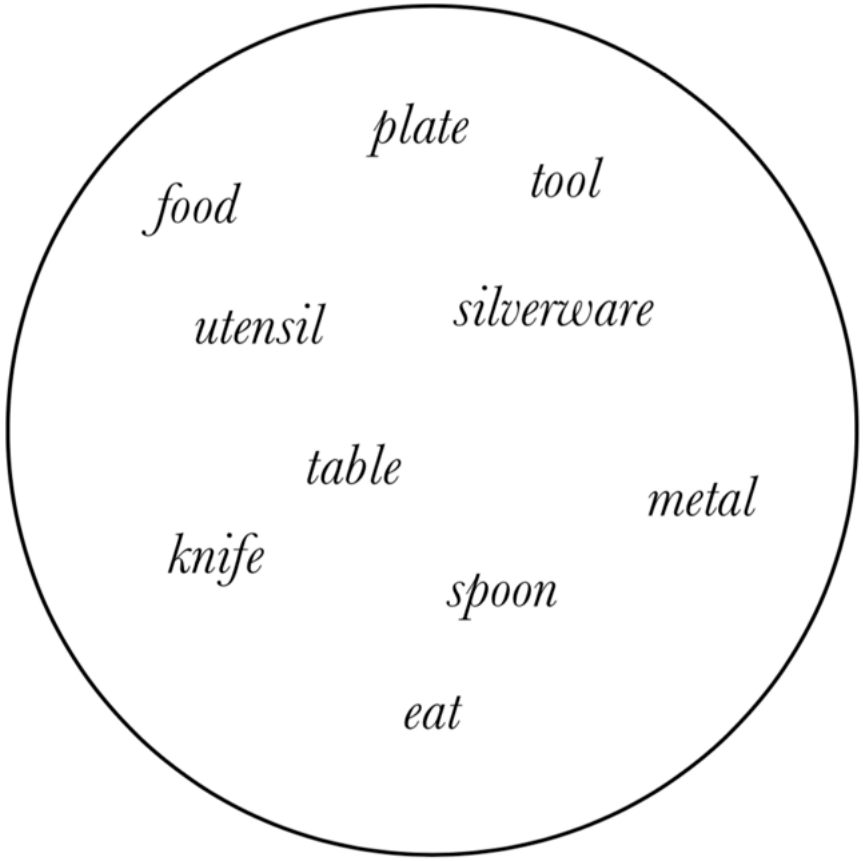
Un termine t è rilevante se una delle seguenti condizioni è verificata:

1. t è un termine di un synset di uno degli elementi contenuti nel vettore NASARI di input



2. oppure se almeno 3 elementi intorno a t (i.e, intorno=1 - direttamente collegati a t -) sono elementi presenti all'interno del vettore NASARI di input

$\mathcal{T}$ =



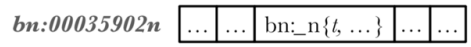
Esempio
passo
2.1

Siamo così riusciti ad ottenere l'insieme dei termini rilevanti (*ora parliamo di termini e non di concetti*)

Esempio *passo* ***2.2***

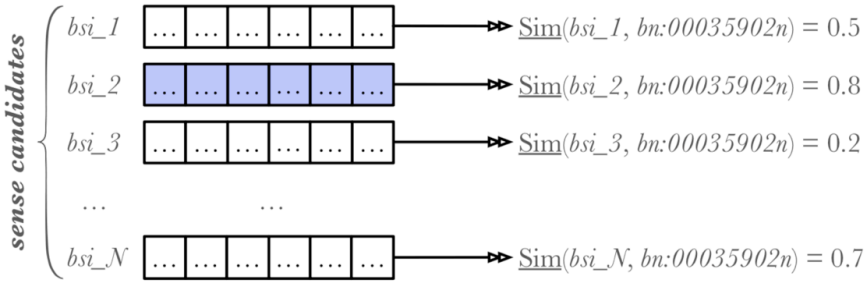
Ora dal nostro insieme di termini rilevanti T dobbiamo passare ad un insieme di concetti rilevanti C (concetti sempre estratti da BabelNet)

A questo punto diventa importante guardare come avevamo ottenuto ognuno di questi termini:



Se un certo termine era direttamente presente nel vettore NASARI (la prima condizione) allora il concetto sarà direttamente presente nel vettore di concetti.

Es. *Banalmente andremo ad inserire in C il BabelSynsetId corrispondente al termine che stavamo valutando*



Se invece il termine era stato inserito tra i termini rilevanti mediante la seconda condizione, recuperiamo tutti i possibili BabelNet synset per quel termine (ovvero i loro vettori NASARI) e calcoliamo la similarità di ognuno di essi con il vettore NASARI di input.

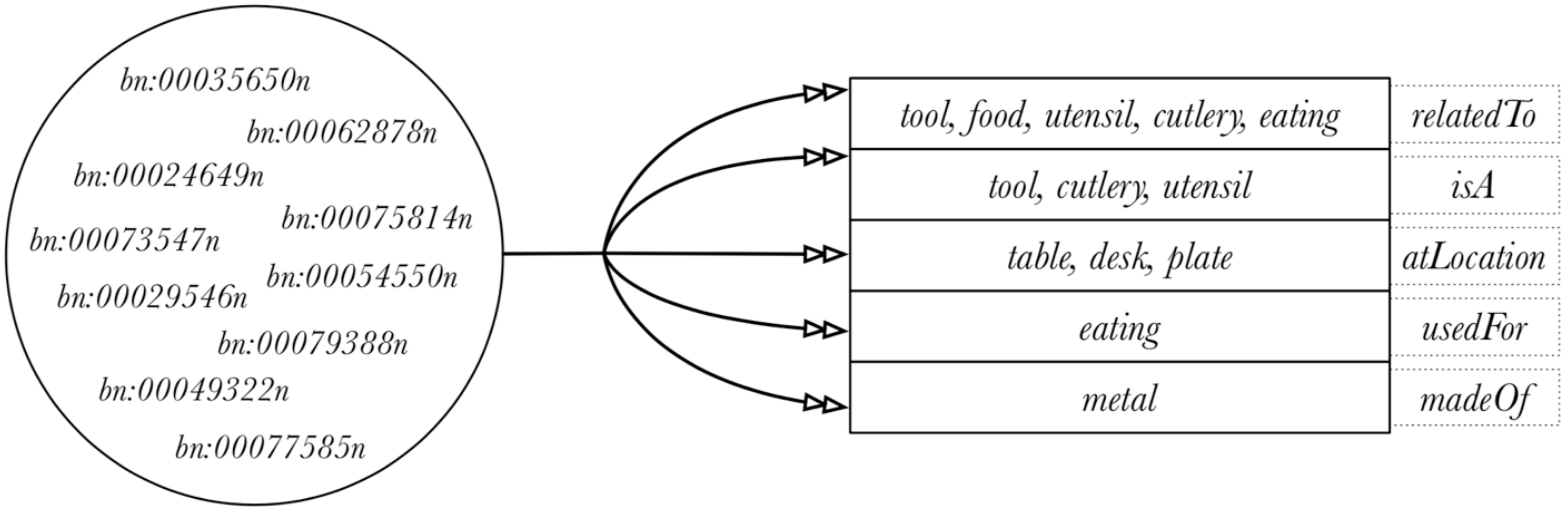
Nota:
Se il valor di similarità del senso più simile è maggiore di una certa soglia, il senso è stato trovato.

Nota:
Si una una soglia minima per cercare di ridurre il numero di fallimenti del procedimento.

Grazie a questo procedimento otteniamo un insieme C di concetti costruiti a partire dall'insieme dei termini T.

Esempio
passo
3

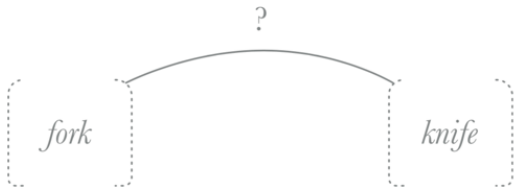
L'ultima fase è quella di inserimento dei concetti nel vettore. Ogni concetto viene inserito nella giusta posizione del vettore tramite le relazioni di ConceptNet



Concept similarity

In tutto questo sembra essere molto importante l'idea di **concept similarity**.

Con questo termine indichiamo il compito di, dati due concetti, assegnare un punteggio di similarità fra i due.



Questo compito è stato risolto da molti software che hanno anche ottenuto punteggi simili a quelli ottenuti da esperti umani.

Inoltre questi software sono molto più d'accordo fra di loro nei punteggi assegnati di quanto lo siano gli esperti umani.

Concept similarity ***Cover***

Per quanto l'algoritmo Cover la similarità si calcola nel modo seguente:

Nota:
L'assunzione generale è che due concetti sono tanto più simili quanto più condividono gli stessi valori sulle stesse dimensioni (e.g., sono tutti e due utilizzati per lo stesso fine)

La similarità tra due vettori (i.e., concetti) è calcolata come la media della similarità calcolata tra ogni dimensione

$$\text{sim}(\vec{c}_i, \vec{c}_j) = \frac{1}{N^*} \cdot \sum_{k=1}^{N^*} \frac{|s_k^i \cap s_k^j|}{\beta\,(\alpha a + (1-\alpha)\;b) + |s_k^i \cap s_k^j|}$$

Per calcolare la similarità tra dimensioni si utilizza un rapporto simmetrico di Tversky, dove:

$$|s_k^i \cap s_k^j|$$

Conta il numero di concetti che sono stati usati come filler per la dimensione d_k nel vettore del concetto c_i e nel vettore del concetto c_j (*intersezione, quelli che compaiono in tutti e due i vettori*)

$$N^*$$

È il numero delle dimensioni presenti in entrambi i vettori

$$a = \min(|s_k^i - s_k^j|, |s_k^j - s_k^i|) \text{ and } b = \max(|s_k^i - s_k^j|, |s_k^j - s_k^i|)$$

α e β fanno tuning sulla mancanza di dati di uno dei due parametri. Questo perché quella mancanza non è dovuta al concetto stesso ma al fatto che si ha meno materiale su quel concetto.

Es. se non so il materiale di una *tenaglia* e quindi non ho il match con quello della *chiave inglese*, la colpa non è della tenaglia ma del fatto che in BabelNet si trovano poche informazioni sulla tenaglia.

Dimension	Dim. Score	V1 | V2 count	Shared values
relatedTo	0.68	44 | 06	*feather, chicken, roosting, vertebrate*
isA	0.58	07 | 04	*animal*

Consideriamo due concetti *cock* e *bird*. Il sistema Cover calcola una similarità di 0.63 contro il *gold standard* di 0.65

Esempio

Per ogni feature (i.e., **dimensione**) in comune, ad esempio “relatedTo”, prendiamo tutti i valori dei vettori in quella colonna (44 nel caso di bird che è una parola più comune e solo 6 nel caso di cock) e cerchiamo gli elementi in comune (in questo caso ne troviamo 4, che sembrano pochi ma sono 4 su 6 che è tanto) e applichiamo la formula precedente e diamo un punteggio a quella features.

Facciamo una media complessiva tra tutte le features trovate e individuiamo un totale di indice di correlazione pari a: **`0.63`**, che è ottimo considerando che il risultato “perfetto”, umano è di 0.65.

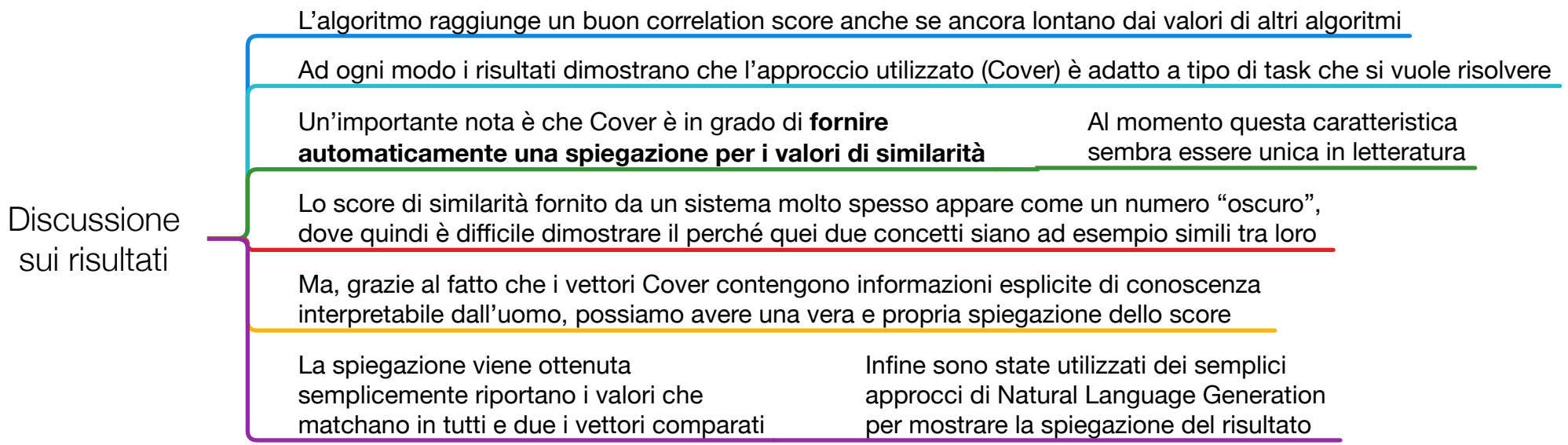
System	RG		MC		WS-Sim		SemEval 2017	
	ρ	r	ρ	r	ρ	r	ρ	r
COVER (Selected data)	0.82	0.88	0.89	**0.91**	0.69	0.70	0.68	0.67
COVER (Full data)	0.76	0.81	0.74	0.79	0.61	0.60	0.65	0.63
NASARI [7,9]	0.88	**0.91**	-	-	-	-	0.68	0.68
ADW [66]	**0.92**	**0.91**	-	-	0.75	0.72	-	-
PPR [1]	0.83	-	**0.92**	-	-	-	-	-
ConceptNet Numberbatch [76]	-	-	-	-	**0.83**	-	-	-
Luminoso [78]	-	-	-	-	-	-	**0.72**	**0.74**
word2vec [49]	0.84	0.83	-	-	0.78	**0.76**	-	-

Risultati

I risultati sono stati calcolati su 4 dataset:

- RG: 65 couples of concepts.
- MC: 28 couples of concepts (subset of the RG dataset).
- WS-Sim: 99 couples of concepts that share some linguistic feature.
- SemEval 2017: 500 couples of words.

The *r* column represents the Pearson correlation, while the *ρ* column indicates the Spearman correlation between our results and the gold standard.


Discussione sui risultati
L’algoritmo raggiunge un buon correlation score anche se ancora lontano dai valori di altri algoritmi
Ad ogni modo i risultati dimostrano che l’approccio utilizzato (Cover) è adatto a tipo di task che si vuole risolvere
Un’importante nota è che Cover è in grado di **fornire automaticamente una spiegazione per i valori di similarità**
Al momento questa caratteristica sembra essere unica in letteratura
Lo score di similarità fornito da un sistema molto spesso appare come un numero “oscuro”, dove quindi è difficile dimostrare il perché quei due concetti siano ad esempio simili tra loro
Ma, grazie al fatto che i vettori Cover contengono informazioni esplicite di conoscenza interpretabile dall’uomo, possiamo avere una vera e propria spiegazione dello score
La spiegazione viene ottenuta semplicemente riportano i valori che matchano in tutti e due i vettori comparati
Infine sono state utilizzati dei semplici approcci di Natural Language Generation per mostrare la spiegazione del risultato

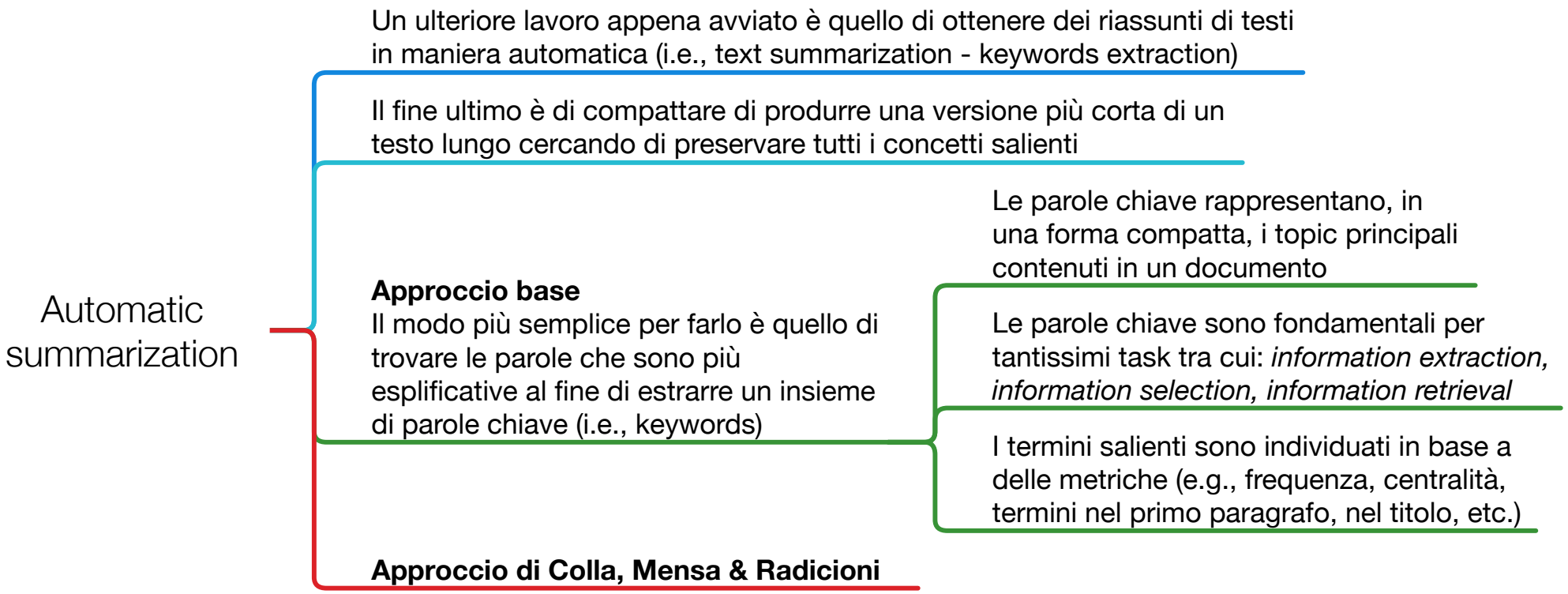
Esempio
Discussione sui risultati

The similarity between *atmosphere* [bn:00006803n] and *ozone* [bn:00060040n] is 2.52 because they are *gas*; they share the same context *chemistry*; they are related to *stratosphere*, *air*, *atmosphere*, *layer*, *ozone*, *atmosphere*, *oxygen*, *gas*.

Es.

Nota:
sono anche stati fatti dei test per asserire la qualità della spiegazione (i.e., descrizione) fornita.

In media lo score è stato di 6.62 (*un valore che è ragionevolmente buono*).


Automatic summarization
Un ulteriore lavoro appena avviato è quello di ottenere dei riassunti di testi in maniera automatica (i.e., text summarization - keywords extraction)
Il fine ultimo è di compattare di produrre una versione più corta di un testo lungo cercando di preservare tutti i concetti salienti
Approccio base
Il modo più semplice per farlo è quello di trovare le parole che sono più esplificative al fine di estrarre un insieme di parole chiave (i.e., keywords)
Le parole chiave rappresentano, in una forma compatta, i topic principali contenuti in un documento
Le parole chiave sono fondamentali per tantissimi task tra cui: information extraction, information selection, information retrieval
I termini salienti sono individuati in base a delle metriche (e.g., frequenza, centralità, termini nel primo paragrafo, nel titolo, etc.)
Approccio di Colla, Mensa & Radicioni

Approccio Colla, Mensa & Radicioni

Questo approccio si basa su un'intuizione: *trovare i concetti salienti invece di contare la frequenza dei termini*

La rilevanza è una quantità di tipo relazionale, non c'è un concetto rilevante di per sé ma è rilevante nel contesto, ovvero in quanto coeso con il titolo del documento

Questa è ovviamente un'assunzione molto forte che ad esempio in un documento privo di titolo non funziona già più

Si procede in due fasi:

1. Fase di disambiguazione
Il titolo e il body di un documento viene disambiguato (questo step permette di tirar via le stop-words e liberarsi di varie varianti morfologiche)

Il documento viene di fatto riscritto in una sequenza di identificatori concettuali

$$c(x) = \frac{1}{|T|} \sum_{y_i \in T} \text{semrel}(x, y_i)$$

2. Fase di matching
Viene calcolata la centralità di ogni concetto nel *body* del documento con il concetto contenuto nel suo *titolo*

La centralità c di un concetto corrispondente al termine x nel body è calcolata come una funzione di *semantic relatedness* rispetto ai concetti nel titolo

Nota:
`semrel` è la vicinanza semantica tra il termine x e il termine del titolo y_i

resource	metrics
NASARI	$\text{semrel}(x, y_i) = \begin{cases} 1 & \text{if } \rho_x^{\vec{y_i}} = 1; \\ 0 & \text{if } x \notin \vec{y_i}; \\ \left(1 - \frac{\rho_x^{\vec{y_i}}}{length(\vec{y_i})}\right) & \text{otherwise.} \end{cases}$
NASARIE	$\text{semrel}(x, y_i) = cosSim(\vec{x}, \vec{y_i})$
COVER	$\text{semrel}(x, y_i) = STRM(\vec{x}, \vec{y_i})$
UCI	$score(w_1, w_2, \epsilon) = \log \frac{p(w_1, w_2, \epsilon)}{p(w_1)p(w_2)}$
UMASS	$score(w_1, w_2, \epsilon) = \log \frac{D(w_1, w_2) + \epsilon}{D(w_2)}$

Metriche di *semantic relatedness*

participant	k	P(%)	R(%)	F(%)
Alch Con	all	16.71	2.81	4.82
Alch Key	all	12.40	16.71	18.24
Calais_Soc	all	13.69	2.60	4.29
KP-Miner	all	40.19	14.46	21.27
Maui	all	27.46	20.30	23.34
TagMe	all	21.02	35.89	26.51
TxtRaz Top	all	6.28	11.52	8.13
Zem Key	all	29.75	5.15	8.78
NASARI	all	39.83	10.86	17.06
NASARIE	all	27.72	36.16	31.38
UCI	all	29.68	**46.28**	**36.17**
UMASS	all	26.76	43.08	33.02
COVER	all	**50.36**	8.49	14.54

Risultati dei test condotti