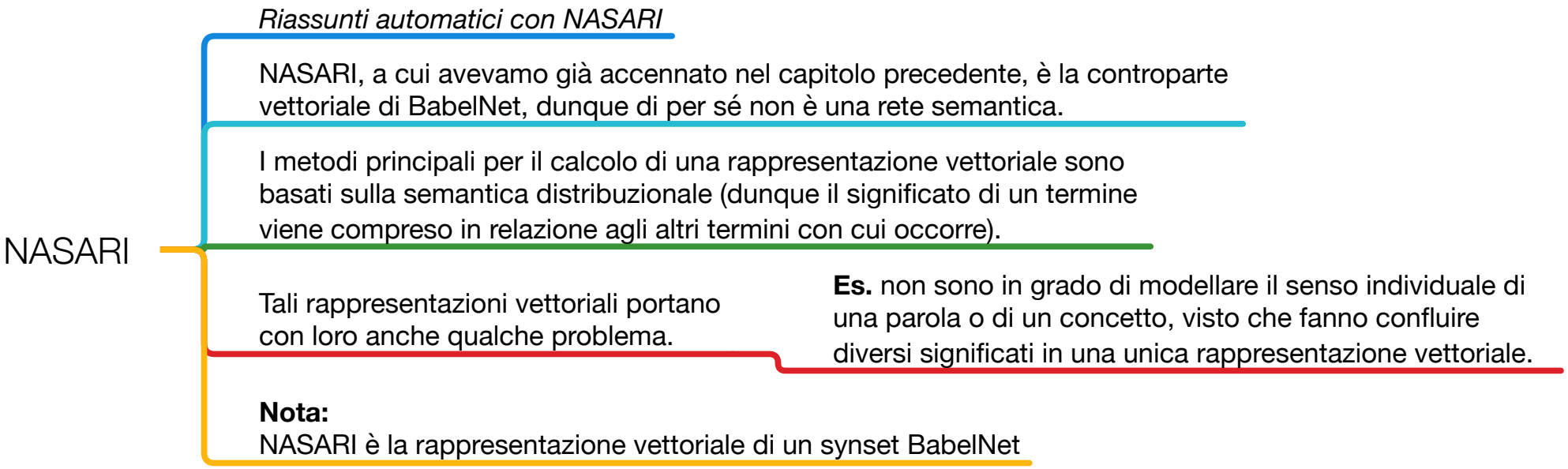
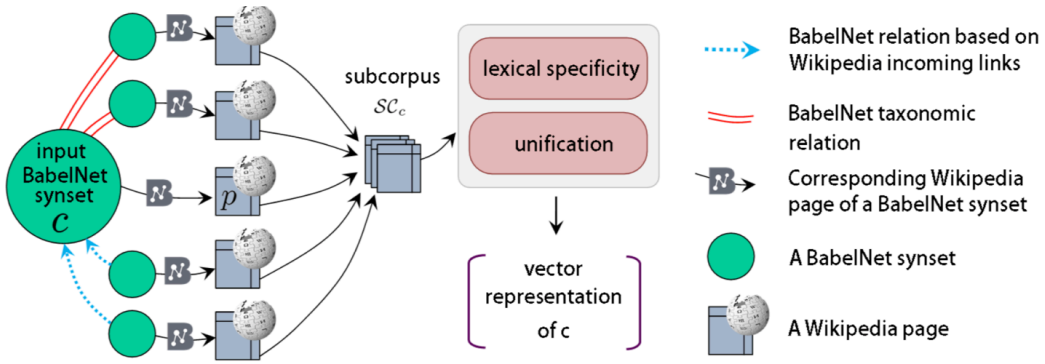

NASARI
Riassunti automatici con NASARI
NASARI, a cui avevamo già accennato nel capitolo precedente, è la controparte vettoriale di BabelNet, dunque di per sé non è una rete semantica.
I metodi principali per il calcolo di una rappresentazione vettoriale sono basati sulla semantica distribuzionale (dunque il significato di un termine viene compreso in relazione agli altri termini con cui occorre).
Tali rappresentazioni vettoriali portano con loro anche qualche problema.
Es. non sono in grado di modellare il senso individuale di una parola o di un concetto, visto che fanno confluire diversi significati in una unica rappresentazione vettoriale.
Nota:
NASARI è la rappresentazione vettoriale di un synset BabelNet

MUFFIN

Una prima soluzione a questo problema viene data dal Multilingual, UniFied and Flexible INterpretation)(**MUFFIN**).

In questo caso non vengono costruiti vettori che contano le co-occorrenze tra parole, bensì le co-occorrenze tra synset di BabelNet.



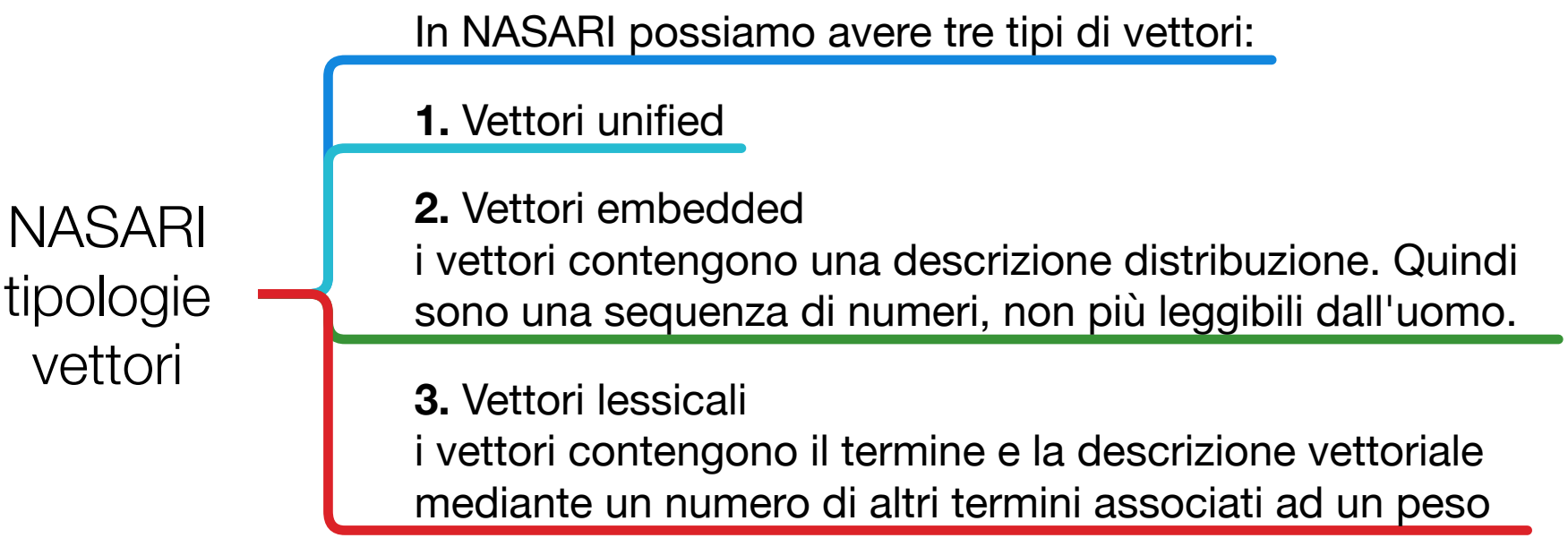
Dato un synset vengono raccolte da Wikipedia le informazioni sul contesto

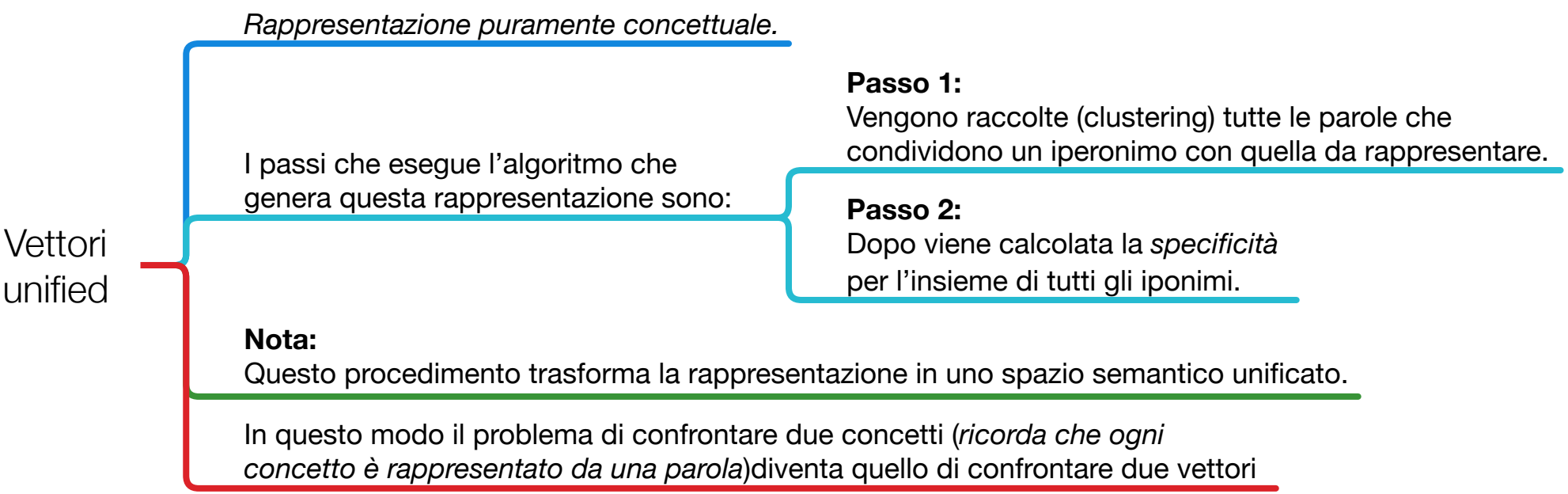
A quel punto analizzando le informazioni sul contesto e comparandole con tutto il corpus di WikiPedia si ottiene una rappresentazione vettoriale per quel determinato synset

Chiaramente una delle problematiche maggiori in questo caso è quella di generare un vettore che rappresenti bene i concetti, **anche nei casi ambigui**!

Es. il vettore per il concetto "piano" deve permetterci di disambiguare tra l'aggettivo e il sostantivo.

Nota:
Come detto l'intero procedimento viene fatto con l'ausilio di Wikipedia.


NASARI
tipologie
vettori
In NASARI possiamo avere tre tipi di vettori:
1. Vettori unified
2. Vettori embedded
i vettori contengono una descrizione distribuzione. Quindi sono una sequenza di numeri, non più leggibili dall'uomo.
3. Vettori lessicali
i vettori contengono il termine e la descrizione vettoriale mediante un numero di altri termini associati ad un peso


Vettori unified
Rappresentazione puramente concettuale.
I passi che esegue l'algoritmo che genera questa rappresentazione sono:
Passo 1:
Vengono raccolte (clustering) tutte le parole che condividono un iperonimo con quella da rappresentare.
Passo 2:
Dopo viene calcolata la *specificità* per l'insieme di tutti gli iponimi.
Nota:
Questo procedimento trasforma la rappresentazione in uno spazio semantico unificato.
In questo modo il problema di confrontare due concetti (*ricorda che ogni concetto è rappresentato da una parola*)diventa quello di confrontare due vettori

Ottenere insieme di concetti

L'obiettivo ora è quello di associare un insieme di concetti (i.e., di BabelNet synsets) C_W={c1,...,cn} con una certa parola w:

Nota:
Ripetiamo ovviamente lo stesso procedimento anche per la seconda parola/concetto da confrontare

1. Se w esiste in BabelNet l'insieme di sensi (i.e., concetti) può essere ottenuto semplicemente prelevando quelli definiti in BabelNet per quella parola

2. Se invece w non esiste in BabelNet possiamo recuperare l'insieme di sensi (i.e., concetti) usando i **piped links**.

I piped link di Wikipedia sono degli hyperlink che compaiono nel corpo di un articolo Wikipedia fornendo un link ad un altro articolo (sempre di Wikipedia).

Es. `[[Crane (Machine)|dockside crane]]`
È un hyperlink che appare nel testo come "dockside crane" ma riporta l'utente ad una pagina Wikipedia dal titolo "Crane (machine)"

Nota by Marco:
Credo quindi che come concetti vengono usate tutte le parole, diverse dal titolo della pagina Wiki, usate per rimandare a quella pagina

Rango

Una volta che abbiamo ottenuto l'insieme C_w dei concetti associati ad una certa parola w, calcoliamo tutte le rappresentazioni vettoriali (i.e., unified vector, quello visto nella MM "Unified Vector") di tutti i concetti all'interno dell'insieme C_w

Nota:
Ripetiamo ovviamente lo stesso procedimento anche per il secondo concetto/parola che stiamo confrontando

Per confrontare due concetti/parole (i.e., vettori) dobbiamo stabilire una metrica da utilizzare.

Una tipica metrica è quella del **rango** (i.e., rank).

$$WO(v_1, v_2) = \frac{\sum_{q \in O} \left(rank(q, v_1) + rank(q, v_2)\right)^{-1}}{\sum_{i=1}^{|O|} (2i)^{-1}}$$

Ora come metodo di comparazione di due vettori usiamo la **sovrapposizione pesata** (Pilehvar et al., 2013) che è un metodo basato sul rank (i.e., ovvero su quanto un termine occorre).

O

O è l'insieme di dimensioni (i.e., caratteristiche/relazioni e.g., isA, hasPart) sovrapposte tra i due vettori

$rank(q, v_i)$

È il rank sulla dimensione q nel vettore v_i

$$sim(w_1, w_2) = \max_{v_1 \in C_{w_1}, v_2 \in C_{w_2}} \sqrt{WO(v_1, v_2)}$$

La similarità tra due parole w_1 e w_2 viene calcolata come la similarità dei due sensi più vicini tra loro

Nota:
Ricorda che ogni vettore in C_{w1} e C_{w2} rappresenta un senso di BabelNet

Nota:
WO = Weighted Overlap (i.e., sovrapposizione pesata)

Ricapitolando

Prendiamo due parole (i.e., concetti) e vogliamo verificare quanto siano simili tra loro (i.e., semanticamente simili)

Per ognuna delle due parole andiamo a creare una insieme C_w di concetti (i.e., BabelNet synsets) associati a quella parola (*il metodo è descritto nella MM "Ottenere insieme di concetti"*)

Una volta che abbiamo i due insiemi di concetti andiamo a rappresentare ogni concetto presente nei due insiemi in forma vettoriale (utilizzando in metodo descritto nella MM "Vettori unified")

A questo punto la similarità tra le due parole (i.e., concetti) sarà data dalla formula:

$$sim(w_1, w_2) = \max_{v_1 \in C_{w_1}, v_2 \in C_{w_2}} \sqrt{WO(v_1, v_2)}$$

La similarità corrisponde quindi al valore di similarità più alto (i.e., max) che abbiamo trovato comparando tutte le combinazioni di vettori prelevati da C_{w1} e C_{w2}

$$WO(v_1, v_2)$$

Nota:
Mi restituisce un valore di similarità, tanto più alto quando i due vettori (i.e., i due BabelNet synset) sono simili (i.e., essendo vettori dovremmo dire "vicini") tra loro