# Activités du Lundi 28 Janvier au Vendredi 1 Janvier :

**Textes lus :**

Chklovski et Pantel. Large-Scale Extraction of Fine-Grained Semantic Relations between Verbs

* Ressources :
  + Wordnet
  + Evca (english verb classes and alternations) 3200 verbes classifiés en 191 classes
* Savoir si un event s’est produit après tel autre (acheté :: vendre)
  + « X buys Y » happens-before « X sells Y »
* Verbe = véhicule primaire pour décrire les events et relations entre entités
* Troponyme : manière de (engloutir, déguster = manière de manger)
* DIRT = algorithme qui apprend automatiquement des expressions paraphrases à partir du texte :
* Is solved by = resolves = find a solution to = deals with … = paraphrases

Bethard, Martin et Klingenstein, (2007). *Timelines from Text: Identification of Syntactic Temporal Relations*

* Paires Verbe + causal
* Utilisation de TimeBank
* SVM model –implémentation TinySVM avec le standard one-vs-rest
* But : produire de la représentation des connaissances en identifiant les composants sémantiques et les intégrer dans un graph
* 895 paires extraites du corpus
* Evaluation 🡪 tester plusieurs petites combinaisons de features : « For example, using just the syntactic path or the tense of the clausal event, models were able to achieve 75% accuracy. Accuracies around 70% were achieved using the part of speech or auxiliaries of the clausal event, or using the words between the two events. »

Chambers et Jurafsky, (2008). *Jointly Combining Implicit Constraints Improves Temporal Ordering*

* Contraintes : transitivité et normalisation de la timex
* Incorporer ces contraintes dans un ILP framework
* Utilisent les relations de TempEval-07 (before, after, vague) sur un document entier plutôt que sur chaque phrase, la tâche inclus tous les event étiquetés de TimeBank + inclusion d’une composante de raisonnement temporel et prendre en compte les contraintes
* Relations e-e et e-t
* **Transitive closure** : La « fermeture transitive » a été la première proposition proposée de ne pas aborder le problème du graphe d'événements connectés, mais plutôt d'élargir la taille des données d'apprentissage pour des relations telles qu'avant. Inclus e-t et t-t pas seulement e-e.
* ILP n’améliore pas leurs résultats, ils ont toujours une exactitude de 66,8%
  + Base : résultats avec TimeBank
  + Global : résultats de ilp avec contraintes utilisant les scores de confiance
  + Global+time : résultats ILP + e-t, t-t
  + ILP n’améliore pas les résultats seuls mais si on ajoute les relations timex ça augmente les résultats de contrainte globale.

Yoshikawa, Riedel, Asahara et Matsumoto, (2009). *Jointly Identifying Temporal Relations with Markov Logic*

* Utilisation d’un modèle de Markov qui identifie conjointement les relations des trois types de relations simultanément
* Markov Logic au lieu de ILP pour deux avantages :
  + if A before B and B overlap with C, then A happens before C
  + trouver les bons features et fournir des données en input + exploiter et comparer les méthodes globales d’inférences
* SRL : Apprentissage relationnel statistique (SRL)

tempsFutur(e) -> nonBeforeDCT(e) : si le temps de l’événement est au futur alors il arrive après le DCT

Markov Logic permet d’en dire plus :

beforeDCT(e1) ^ nonBeforeDCT(e2) -> before(e1,e2) : si e1 est avant le DCT et e2 est après le DCT alors e1 est par conséquent avant e2

* Exemple de loi de transition
* Cette loi est le cœur de l’idée de leur approche jointe

**Markov Logic Network (formule globale) :**

3 prédicats cachés :

* relE2T(e, t, r) : relation temporelle de la classe **r** entre un event **e** et le timex **t 🡪 r(e, t) - A**
* relDCT(e, r) : relation temporelle **r** entre un event **e** et le **DCT 🡪 r(e, DCT) - B**
* relE2E(e1, e2, r) : relation **r** entre 2 event **e1 et e2** de phrases adjacentes **🡪 r(e1, e2) - C**

2 prédicats observables :

* relT2T(t1, t2, r) : relation **r** entre 2 timex **t1 et t2 🡪 r(t1, t2)**
* dctOrder(t, r) : relation **r** entre un timex **t** et un **DCT** fixe **🡪 r(t, DCT)**

Résultats :

* La formule globale n’améliore pas que les scores stricts mais aussi les scores relaxed pour toutes les tâches. Production d’étiquettes plus ambiguës dans les cas où le modèle local a été trop confiant (before-or-overlap au lieu de before ou overlap) et réduit les erreurs = plus cohérent
* Les relations vagues sont cohérentes mais peu informatives
* L’apprentissage de paramètres fiables sont + difficiles lorsqu’on a des données bruitées.

Mirza et Tonelli, (2014). *Classifying Temporal Relations with Simple Features*

* Pour obtenir les features dont A-N. Minard et P. Paramita se sont inspirées

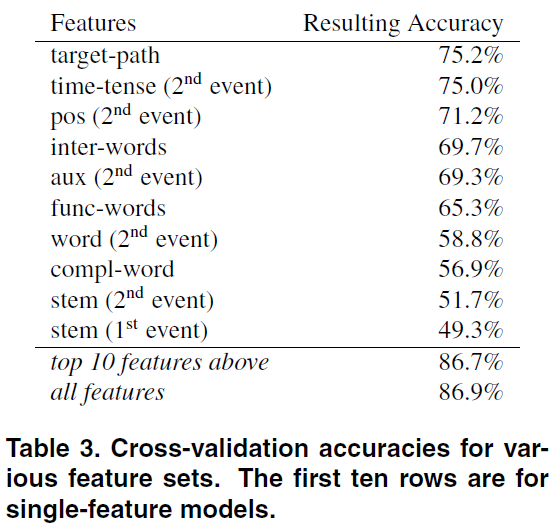
Chambers, Wang, Jurafsky, (2007). *Classifying Temporal Relations Between Events*

* Amelioration des résultats de travaux déjà présentés grâce à de nouveaux features

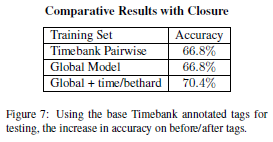
|  |  |  |
| --- | --- | --- |
| **Auteurs** | **Features utilisés** | **Observations** |
| Bethard, Martin et Klingenstein, (2007). *Timelines from Text: Identification of Syntactic Temporal Relations* | 2 ensembles de features :   * Description linguistique d’un event isolé * **Word** * **Pos** : the **Penn TreeBank gold-standard pos label for the event** * **Stem** -> = lemme ? non le stem coupe le mot mangeais -> mang et le lemme nous remet à la forme canonique. Pour l’anglais le stem suffit peut être.   Moins précis mais peut éviter des erreurs car c’est juste du découpage   * **Aux** : auxiliaires qui modifient l’event sont inclus * **Modal** : pour les modaux * **Time-class** : TimeBank gold-standard class label for event (state or reporting) * **Time-pos** : TimeBank gold-standard pos pour les events * **Time-tense** : TimeBank gold-standard temps pour les events * **Time-aspect** : TimeBank gold-standard aspect pour les events (progressive / perfective) * **Time-polarity** : TimeBank gold-standard polarité pour les events (pos / neg) * Comment les mots sont connectés avec les events (relations) * **Compl-word** : la forme textuelle du complémentateur 🡪 un complémentateur est un mot utilisé pour introduire une clause de complément (to, that, because) * **Compl-type** : Le type du complémentateur (after, overlap) * **Target-path** : le chemin syntaxique partant de la clause jusqu’à sa tête. * **Inter-words** : tous les mots entre deux events * **Func-words** : comme inter-words mais avec :   + Prepositions:  of, at, in, without, between   + Pronouns:  he, they, anybody, it, one   + Determiners:  the, a, that, my, more, much, either, neither   + Conjunctions:  and, that, when, while, although, or   + Auxiliary:  verbs be (is, am, are), have, got, do   + Particles:  no, not, nor, as | Résultats :  SVM obtient 89,2% d’exactitude.  Si un des deux events est « None » 🡪 fait baisser l’exactitude  La combinaison des 10 meilleurs features = 86,7% d’accuracy, tous les features ensemble donnent un gain de 0,2%. Cela signifie que seulement les 10 meilleurs features suffisent à donner un bon résultat.  Sur 10 features :   * 4 pour le mot * 5 pour le causal * 1 pour le verbe * Signifie que le causal joue un rôle + important que le verbe dans la relation temporelle.   En prévisions futures : identifier les features utiles  Ils essaient des petites combinaisons pour voir lesquels apportent les meilleurs résultats 🡪 Features engineering |
| Chambers et Jurafsky, (2008). *Jointly Combining Implicit Constraints Improves Temporal Ordering* | **Features de chaque event :**  **Word** : la forme textuelle des deux events  **Lemma** : le lemme des deux events  **Synset** : ensemble de synonymes issus de WordNet des deux events  **Pos** : « 4 POS tags, 3 before, 1 event » pour les 2 events ? -> 3 étiquettes des 3 mots d’avant et l’étiquette de l’event  **POS** **bigram** : le pos des bigrams de l’event et ses tags précédents (des deux events) (juste celui d’avant, et l’actuel en un seul trait)  **Prep** : si il y a une préposition dans chaque event  **Tense** : le temps de chaque event  **Aspect** : l’aspect de chaque event (grammatical)  **Modal** : la modalité de chaque event  **Polarity** : pos / neg  **Class** : la classe aspectuelle de chaque event  **Features de chaque paire :**  **Tense pair** : les 2 temps concaténés  **Aspect pair** : les 2 aspects concaténés  **Class** **pair** : les 2 classes concaténées  **POS** **pair** : les 2 pos concaténés  **Tense** **match** : booléen qui renvoie True si les temps des deux events sont les mêmes  **Aspect** **match** : booléen qui renvoie True si les aspects des deux events sont les mêmes  **Classe** **match** : booléen qui renvoie True si les classes des deux events sont les mêmes  **Dominates** : booléen qui renvoie True si l’event 1 domine l’event 2 au niveau syntaxique  **Text** **order** : booléen qui renvoie True si le premier event apparait en premier dans le document  **Entity** **match** : booléen qui renvoie True si les deux events partagent une entité comme argument  **Same** **sent** : booléen qui renvoie True si les deux event apparaissent dans la même phrase    **Transitive closure** : La « fermeture transitive » a été la première proposition proposée de ne pas aborder le problème du graphe d'événements connectés, mais plutôt d'élargir la taille des données d'apprentissage pour des relations telles qu'avant. Inclus e-t et t-t pas seulement e-e. | **Quelle est la différence entre l’aspect et la classe qui fait référence à la classe aspectuelle ?**  Dans TimeMl il y a plusieurs classes et y en a une qui est aspectuelle (begin, start, restart, stop) -> lié à la sémantique  Et l’aspect, perfect, progressif, etc. -> lié à la grammaire   * Dominant selon le chemin de l’arbre syntaxique   **s’ils partagent au moins un argument ?**  entité partagée par un argument : sujet du verbe est un agent et ça peut être l’objet du premier événement ou du deuxième  Marie continue d’aller au sport. Continue et aller partagent la même entité Marie qui participé à cet événement.  Lieu, instrument… |
| Yoshikawa, Riedel, Asahara et Matsumoto, (2009). *Jointly Identifying Temporal Relations with Markov Logic* | relE2T(e, t, r) : relation temporelle de la classe **r** entre un event **e** et le timex **t 🡪 r(e, t) - A**  EVENT-word  EVENT-POS  EVENT-stem  EVENT-aspect  EVENT-tense  EVENT-class  EVENT-polarity  TIMEX3-word  TIMEX3-POS : si c’est un numeral etc  TIMEX3-value : valeur normalisée  TIMEX3-type : si c’est date, durée, duration, time, set  TIME-DCT order : selon évènement est ce que l’event est after / before le DCT  Positional order : ordre entre événement et timex  In/outside : ???  Unigram(word) : la forme textuelle de l’event  Unigram(POS) : le POS de l’event  Bigram(POS) : (posEvent posEvent+1)  Trigram(POS) : (posEvent posEvent+1 posEvent+2)  Dependency-Word : si relation type sujet : on prend la forme textuelle de la relation ??  Dependency-POS : pos du mot avec laquelle ils ont une dépendance ??  relDCT(e, r) : relation temporelle **r** entre un event **e** et le **DCT 🡪 r(e, DCT) - B**  EVENT-aspect  EVENT-tense  EVENT-class  TIME-DCT order  Dependency-Word  Dependency-POS  relE2E(e1, e2, r) : relation **r** entre 2 event **e1 et e2** de phrases adjacentes **🡪 r(e1, e2) - C**  EVENT-word  EVENT-POS  EVENT-stem  EVENT-aspect  EVENT-tense  EVENT-class  EVENT-polarity  Unigram(word)  Unigram(POS)  Trigram(POS)  Dependency-Word | Résultats : Le model global n’améliore pas que les scores stricts mais aussi les scores relaxed pour toutes les tâches. Production d’étiquettes plus ambiguës dans les cas où le modèle local a été trop confiant (before-or-overlap au lieu de before ou overlap) et réduit les erreurs = plus cohérent  Le modèle obtient les + hauts scores dans 2 tâches /3 |
| Mirza et Tonelli, (2014). *Classifying Temporal Relations with Simple Features* | **Chaine** : chaine des tokens et lemmes de e1 et e2  **Descripteurs grammaticaux** : PoS tag, chunk,et binaire pour dire si les events ont le même PoS  **Contexte textuel** : la distance en mots entre e1 et e2  **Attributs d’entités** (attributs des événements) : Classe / temps / aspect / polarité  **Informations sur les dépendances** : selon le chemin de dépendances (arbre syntaxique), est-ce que e1 ou e2 est le verbe racine.  **Signaux temporels** : la position des événements, exemple : before/after e1, before/after e2  **Connecteurs temporels discursifs** : pertinent que pour les paires event/event |  |
| Chambers, Wang, Jurafsky, (2007). *Classifying Temporal Relations Between Events* | * Attributs des events inspirés des travaux de Pustejovsky et al., 2003:   + **Tense** (none, present, past, future)   + **Grammatical aspect** (none, prog, perfect, prog\_perfect)   + **Modality** (none, to, should, would, could, can, might)   + **Polarity** (positive, negative)   + **Event class** (report, aspectual, state, I\_state, I\_action, perception, occurrence) * Features pour l’apprentissage automatique :   + **PoS tags** (2 before event)   + **Lemmas**   + **WordNet synsets**   + **Auxiliaries** before event (derivation de be et have)   + **Modals** before event (may, might, etc)   Ils ont essayé différents ensembles de features pour chacun des 5 attributs. Les features qui ont été sélectionnés :   * **tense** : POS-2-event, POS-1-event, POS-of-event, have\_word, be\_word * **aspect** : POS-of-event, modal\_word, be\_word * **class** : synset * **modality** : none * **polarity** : none * Attributs Event-Event features   + Eventspecific :     - Booléen même temps     - Booléen même aspect     - Booléen même event class     - Bigram temps (present past -> e1 present e2 past)   + PoS :     - Inclus de Penn Treebank POS tag     - Utilisation de Stanford Parser pour extraire les POS     - Bigram pos event et token avant l’event     - Bigram de e1 et e2   + E-E propriétés syntaxiques :     - Dominance syntaxique     - Indicateur before pour l’ordre des events     - Est-ce que les events sont dans la même phrase   + Phrase prépositionnelle     - Indicateur si l’event appartient à une PP     - Event dominance   + Discours temporel   Exemple des features : (lemma1: require) (lemma2: compromise) (dominates: yes) (tense-bigram: past-none) (aspect-bigram: none-none) (tense- match: no) (aspect-match: yes) (before: yes) (same-sent: yes) | Résultats   * 60,45% gold / 59,13% auto / 59,43% auto-split * Meilleurs résultats que les autres travaux grâce aux nouveaux features qu’ils ont ajoutés |

**Résultats obtenus :**

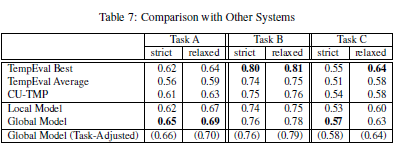
**Bethard, Martin, Klinenstein :**



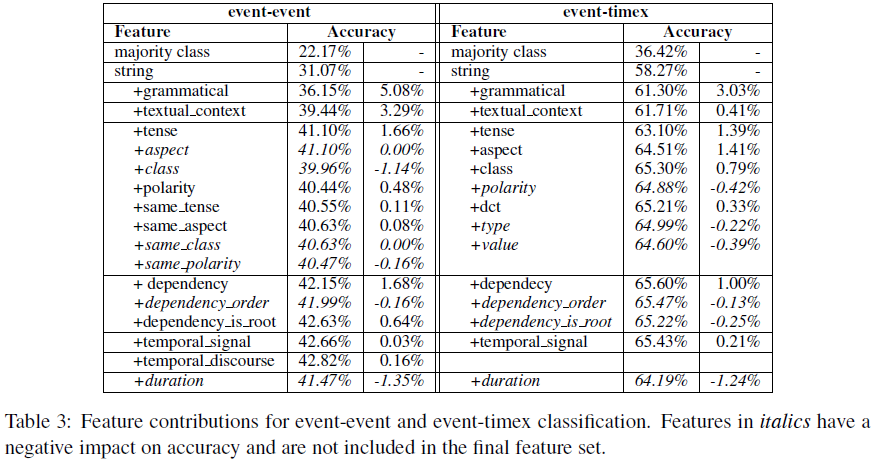
**Chambers, Jurafsky :**



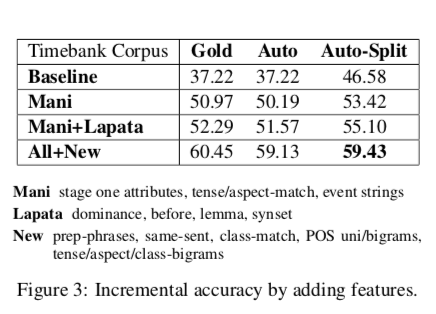
**Yoshikawa, Riedel, Matsumoto :**



**Mirza, Tonelli :**



Chambers, Wang, Jurafsky, (2007). *Classifying Temporal Relations Between Events*



Autres textes sur les features / relations temporelles :

Ha, Baikadi, Licata, Lester, (2010). *NCSU: Modeling Temporal Relations with Markov Logic and Lexical Ontology*

Mirroshandel, Ghassem-Sani, (2012). *Towards Unsupervised Learning of Temporal Relations between Events*

Qu’est-ce que l’on peut en déduire ? Qu’il y a des features qui sont récurrents et essentiels :

* **Mot**\* (tous les auteurs sauf Chambers, Wang, Jurafsky)
  + Bethard: forme textuelle de l’event
  + Chambers et Jurafsky : forme textuelle des deux events
  + Yoshikawa :
    - EVENT-word
    - TIMEX-word
    - Unigram(word)
  + Chambers, Wang, Jurafsky : pas de mot 🡪 donnent directement les lemmes des events
  + Mirza et Tonelli : forme textuelle des deux events
* **PoS**\* (tous les auteurs)
  + Bethard :
    - the Penn TreeBank gold-standard pos label for the event
    - Time-pos
  + Chambers et Jurafsky :
    - « 4 POS tags, 3 before, 1 event » pour les 2 events ? -> 3 étiquettes des 3 mots d’avant et l’étiquette de l’event
    - POS bigram : le pos des bigrams de l’event et ses tags précédents (des deux events) (juste celui d’avant, et l’actuel en un seul trait)
    - POS pair : les 2 pos concaténés
  + Yoshikawa : EVENT-POS, TIMEX3-POS, unigram POS, bigram POS, trigram POS, Dependency-POS
  + Chambers, Wang, Jurafsky : E-E features
    - Penn Treebank POS tag
    - Utilisation de Stanford Parser pour extraire les POS
    - Bigram pos event et token avant l’event
    - Bigram POS de e1 et e2
  + Mirza et Tonelli : POS tag de chaque event
* **Lemme ou stem** (choisir l’un ou l’autre)\*
  + Lemme (Chambers et Jurafsky - Mirza et Tonelli - Chambers, Wang, Jurafsky)
  + Stem (Bethard - Yoshikawa)

Comme nous traitons la temporalité, il semble également essentiel d’inclure des features liés au/à :

* **Temps**\* (présent, passé, futur, none) (tous les auteurs)
  + Bethard :
    - **Time-tense** : TimeBank gold-standard temps pour les event
  + Chambers et Jurafsky :
    - **Tense** : le temps de chaque event
    - **Tense pair** : les 2 temps concaténés
  + Yoshikawa :
    - EVENT-tense
  + Chambers, Wang, Jurafsky :
    - **tense** : POS-2-event, POS-1-event, POS-of-event, have\_word, be\_word
  + Mirza et Tonelli :
    - Attribut d’event : temps
* **Aspect**\* (tous les auteurs)
  + simple (i wash),
  + perfect (i have washed),
  + progressif (i am washing),
  + perfect progressive (i have been washing))
  + none
  + Bethard
    - **Time-aspect** : TimeBank gold-standard aspect pour les event (progressive / perfective)
  + Chambers et Jurafsky :
    - **Aspect** : l’aspect de chaque event (grammatical)
    - **Aspect pair** : les 2 aspects concaténés
  + Yoshikawa :
    - EVENT-aspect
  + Chambers, Wang, Jurafsky :
    - **aspect** : POS-of-event, modal\_word, be\_word
  + Mirza et Tonelli :
    - Attributs d’event : aspect
* **Classe aspectuelle**\* (exemples : begin, start, restart, stop, none)
  + Chambers et Jurafsky :
    - **Class** : la classe aspectuelle de chaque event
* **Signaux temporels**+ : la position des événements : e1 before/after, e2 before/after
  + Mirza et Tonelli :
    - **Signaux temporels** : la position des événements, exemple : before/after e1, before/after e2 (event-event / event-timex)
  + Chambers, Wang, Jurafsky :
    - Indicateur before pour l’ordre des events

La prise en compte de l’environnement est intéressante :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entourage lexical / syntaxique | | | | | |
| Bethard | Chambers et Jurafsky | Yoshikawa | Chambers, Wang, Jurafsky | Mirza et Tonelli | Nos propositions |
| **Inter-words** : tous les mots entre deux events | **Pos** : « 4 POS tags, 3 before, 1 event » pour les 2 events ? -> 3 étiquettes des 3 mots d’avant et l’étiquette de l’event | Bigram(POS) : (posEvent posEvent+1) |  | **Contexte textuel** : la distance en mots entre e1 et e2 | 4 tokens avant l’event   * Aux * Adv * Str * Marqueurs de négation |
| **Func-words** : comme inter-words mais inclus uniquement :  Prepositions:  of, at, in, without, between  Pronouns:  he, they, anybody, it, one  Determiners:  the, a, that, my, more, much, either, neither  Conjunctions:  and, that, when, while, although, or  Auxiliary:  verbs be (is, am, are), have, got, do  Particles:  no, not, nor, as |  | Trigram(POS) : (posEvent posEvent+1 posEvent+2) |  |  | 4 tokens après l’event   * Aux * Adv * Str * Marqueurs de négation |

Nous pouvons inclure des booléens pour les paires :

* Mêmes mots+ ?
* Mêmes temps+ ?
* Mêmes aspects+ ?
* Inclus dans le même syntagme+ ? = chunking ?
  + Bethard : pas de booléen
  + Chambers et Jurafsky :
    - **Tense** **match** : booléen qui renvoie True si les temps des deux events sont les mêmes
    - **Aspect** **match** : booléen qui renvoie True si les aspects des deux events sont les mêmes
    - **Classe** **match** : booléen qui renvoie True si les classes des deux events sont les mêmes
    - **Dominates** : booléen qui renvoie True si l’event 1 domine l’event 2 au niveau syntaxique
    - **Text** **order** : booléen qui renvoie True si le premier event apparait en premier dans le document
    - **Entity** **match** : booléen qui renvoie True si les deux events partagent une entité comme argument
    - **Same** **sent** : booléen qui renvoie True si les deux events apparaissent dans la même phrase
  + Yoshikawa : pas de booléen
  + Mirza et Tonelli :
    - events ont le même PoS ?
  + Chambers, Wang, Jurafsky
    - Est-ce que les events sont dans la même phrase
    - Booléen même temps
    - Booléen même aspect
    - Booléen même event class

Autres :

* **Polarité**\* (positive / négative) (tous les auteurs)

Utilisation de WordNet pour les synsets (synonymes) ? (Chambers, Jurafsky / Chambers, Wang, Jurafsky)

\* : pour chaque événement

+ : pour chaque paire

**Idée** (peut-être mauvaise, à voir)

|  |  |
| --- | --- |
| Feature : synonymes  Extraction de synonymes avec WordNet ? | Inclusion de synonymes des verbes (avec WordNet) pour essayer de trouver dans VerbOcean un équivalent en synonyme ?  Si l’event n’est pas dans VerbOcean, est ce qu’un de ses synonymes s’y trouve ? si oui on prend la relation  xRy ssi match VerbOcean ou ssi synonyme match dans VerbOcean |