Multilingual FrameNet

Vancouver BC Meeting après ACL 2017

Aug 5, 2017

Outline

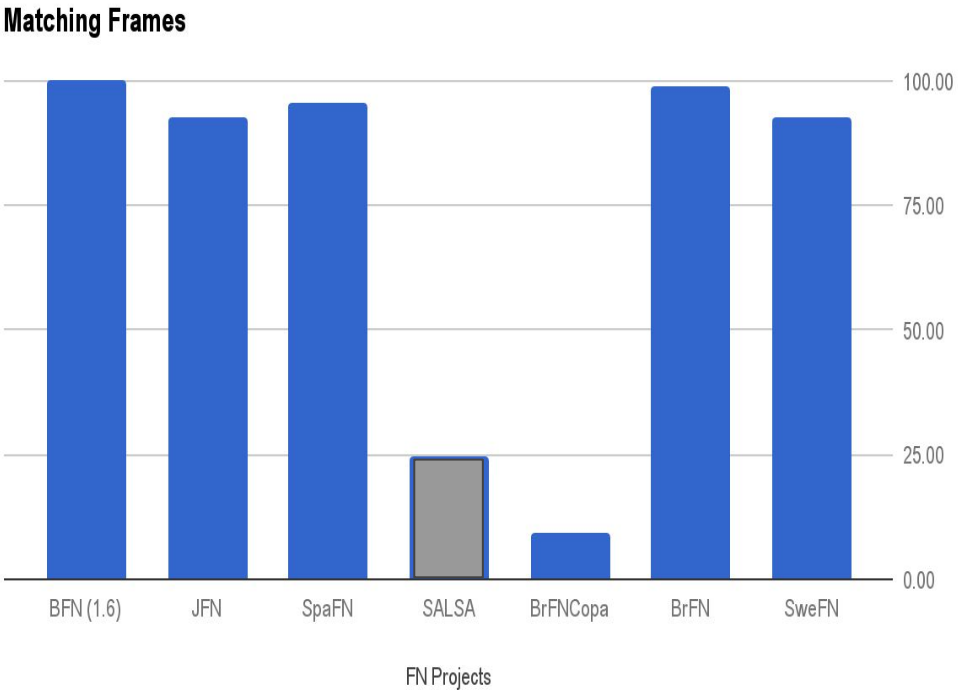
Aligning FrameNet Projects

- Current status
  - ICSI FN vs. the others
  - degree of overlap
    - graph matching algorithm
    - manual vs. automatic correction of alignment
- Practical aspects
  - Tools
    - Restructuring
    - Versioning
  - Maintenance and growth of MLFN

The current state: Matching Frames

Stats:

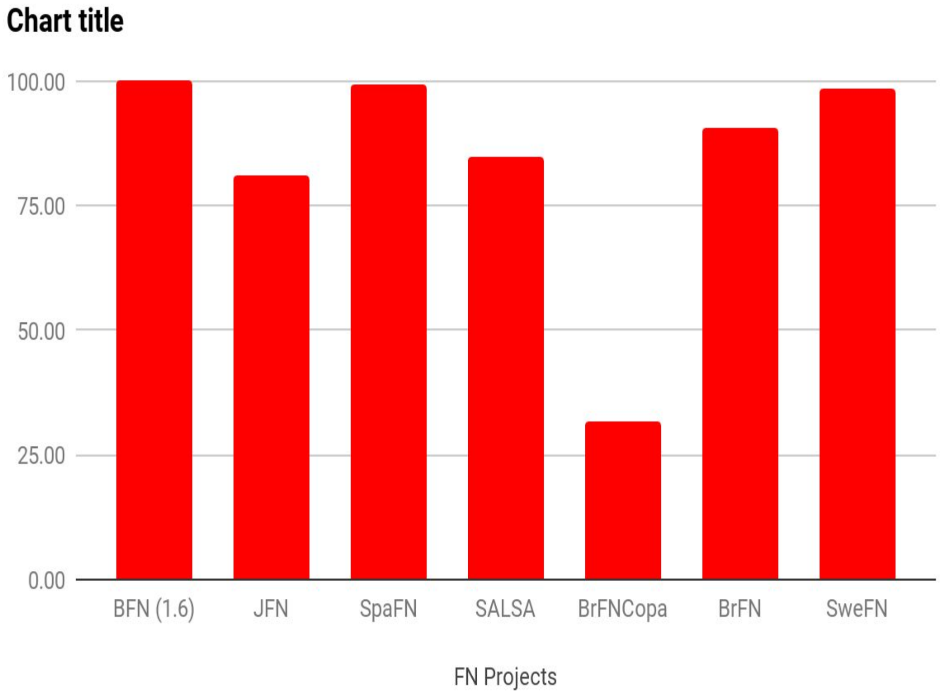
- The rosy picture: relative coverage
  - That is, ratio of frames covered in the various FN projects (w.r.t. Berkeley FN)
  - It might seem that there’s a relatively good overlap, in general
    - Exceptions: SALSA and FN Br/Copa



The current state: Relative Sizes

Stats:

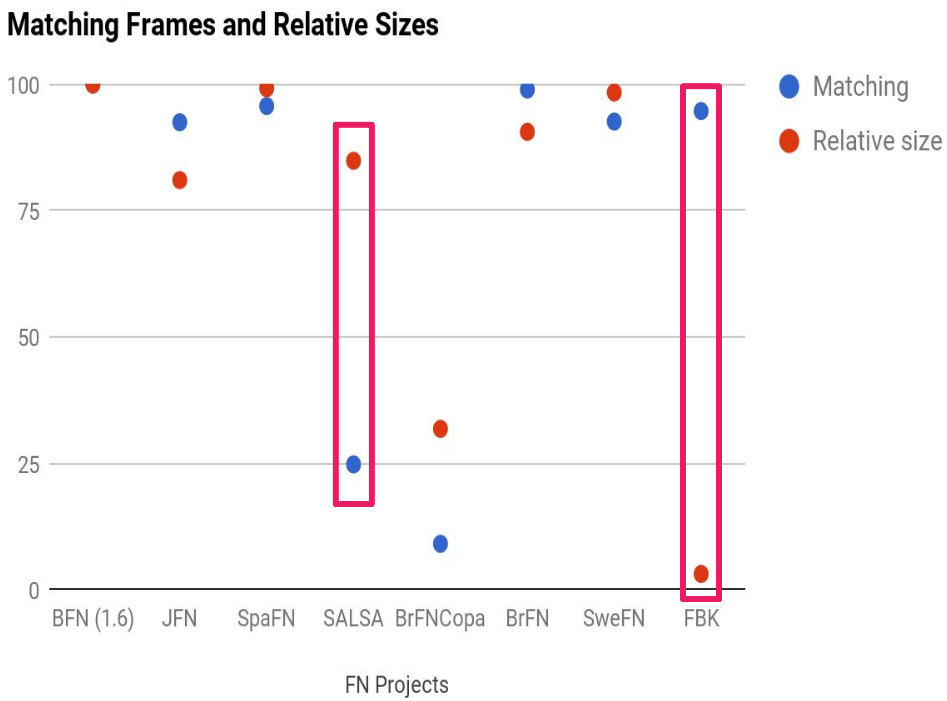
- Here instead the size ratios, still w.r.t. Berkeley FN
    - Some projects cover only a few percent of the frames in BFN



The current state: Compare and Contrast

Stats:

- Under the surface
  - Some projects include only a small part of the frames contained in BFN
- More importantly:
  - **considerable differences**
    - In terms of **LU**s
    - And in terms of **annotated data**



The Current State: More fine-grained differences

- We can divide the non-English FrameNets in two coarse “classes”
    - The ones directly derived from some version of EnFN, but only extended Frames/FEs in a (relatively) limited way
        - These used EnFN Frames as “templates”
            - and filled in LUs and Annotations
                - SpaFN
                - JFN
                - BrFN(Copa)
        - The ones that diverged a lot more,
            - creating a number of new (Proto)Frames
                - SALSA
                - SweFN

The Current State: More fine-grained differences

Which seems to create two different sets problems:

- In the first case:
    - We can rely on BFN’s elements and IDs, and,
        - for each pair of (BFN, xFN):
            - Compare the single Lexical Units for each Frame
            - Compare Frames, FEs, SemTypes and Relations
            - Come up with a metric to assess the similarity
                - Along the lines of the Jaccard Index
- In the second, we cannot; so we either
    - Assume no overlap with any Frame in BFN
    - Or find in BNF the closest matching Frame
        - Which assumes that we already have a reliable mapping among all the overlapping frames

The Current State: More fine-grained differences

- Further problem:
  - The different projects branched off from different versions of BFN
  - Some from FN 1.5 (Spanish, Korean), some from even earlier versions (FN 1.2)
- Thus, even if we limit ourselves to the first class,
  - we now have *two subproblems*:
    - Find a mapping from the current BNF to the BNF version used by the project at hand (let’s call it xFN)
    - Find a mapping from the earlier BNF version to xFN
- Finally, compose the two mappings

The Current State: More fine-grained differences

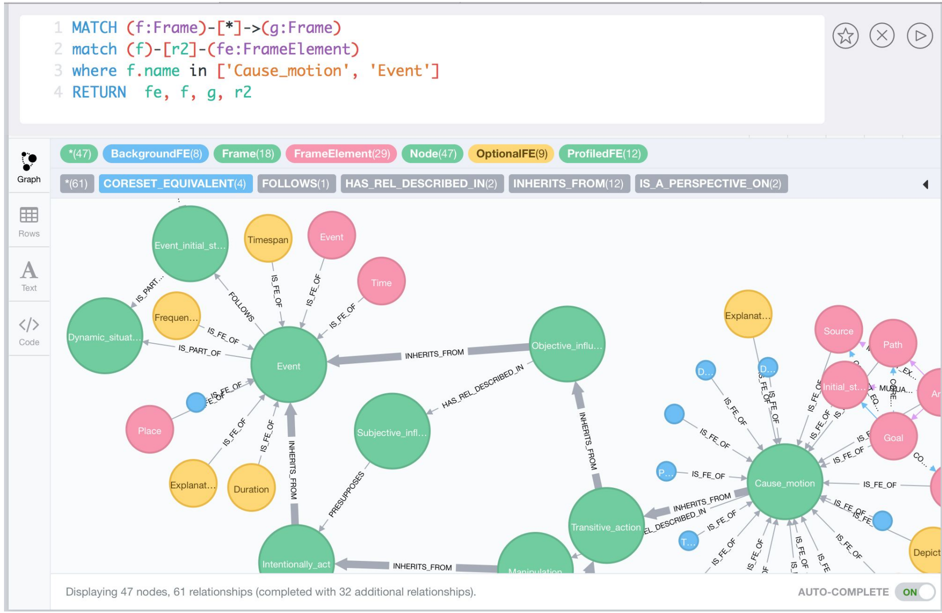
- Interaction of BFN with other languages:
    - From the history of some projects (e.g. SALSA) we know that:
        - Project xFN starts annotating, and adds Frames
        - Some of which, later, *in parallel*, get added to BFN
            - Not exactly the same Frames, just very similar
        - Those BFN Frames then get altered to accommodate differences
        - Some other xFN Frames prompt creation of new BFN Frames
- *BFN's geography has been shifting*
    - Which is true of the other FN Projects' as well
    - And so will MLFN's — arguably to a greater extent
- Bottom line:

  We need the ability to track restructuring as this tectonic shift takes place

The Current State: FN as a Graph

Which brings us to the first step:

- We’ve settled for a graph DB
  - Some projects (BrFN) have taken relations to the next level
  - Restructuring is easier
- FN matching as graph matching
  - We want to exploit a host of graph algorithms
  - Much easier to implement than on a relational DB



```
MATCH (f:Frame)-[*]->(g:Frame)
match (f)-[r2]-(fe:FrameElement)
where f.name in ['Cause_motion', 'Event']
RETURN  fe, f, g, r2
```

Displaying 47 nodes, 61 relationships (completed with 32 additional relationships).

The Current State: the General Strategy

- We start by matching projects
    - Pairwise, i.e. we compare each project with BFN, and
    - For each matching pair, we evaluate the overlap
        - But how?
            - Now we have a new problem

- We might find hints to a solution by looking at the **BRM**™

The Current State: the General Strategy — BRM™

- The **Berkeley Recommended Method**, or **BRM**™
    - Accurate time of genesis unknown (to me), but plausibly around the SALSA project era
    - Used (or recommended) to avoid duplicate frame creation in other languages
    - *In practice*:
        - For each Frame in project xFN
            - Make a list of words in it
            - Translate them
            - And make sure that no BFN Frame contains them
- **Q**: *Can we operationalize this and scale it*
    - *out? (to more xFNs)*
    - *up? (to more data)?*

The Current State: the General Strategy — BRM™

- **Q**: *Can we operationalize this and scale it*
  - *out? (to more xFNs)*
  - *up? (to more data)?*
- **A**: Most likely, yes!
  - In different ways, with different degrees of sophistication
- A simple one: dictionaries?
  - We can use one of the many lexical resources available
  - Including Open Multilingual Wordnet (and similar)
    - We could try to include hypernyms and, if we’re careful, synonyms
- But we are FrameNetters
  - Se we care about the syntactic and semantic environments in which WFs are used
- **Q**: *Can we do better than that?*

The Current State: the General Strategy — Problems

- **Q**: Can we *try* to include syntactic and semantic environments?
- **A**: Well, sort of.

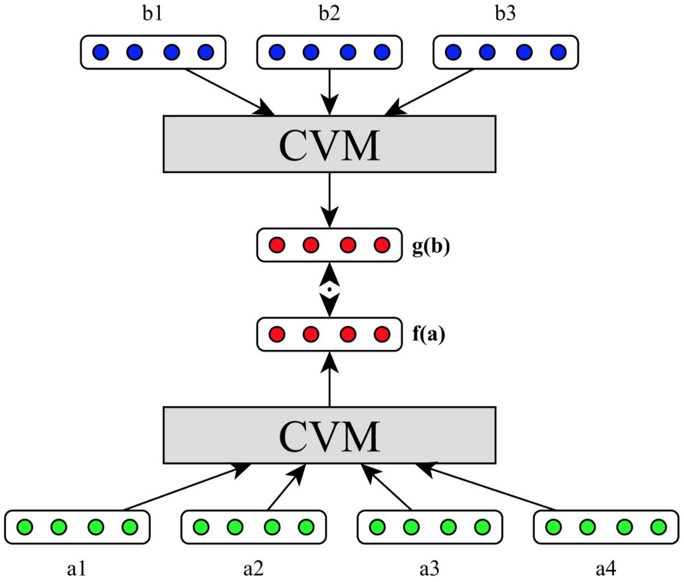
  We could try to use (some form of) distributional word representation
  - Which take *word embeddings* into consideration, and
    - map those onto linear spaces (Word2vec, GloVe)
    - These methods look at *word windows* of a few words
    - Syntactic relations do not matter
  - There are more sophisticated methods that do look at them
    - (Pado and Lapata 2007)
- **But all these leave us with another subtask**
  - Now we have to align vector representations!

The Current State: the General Strategy — Problems

- **Q**: Can we *try* to include syntactic and semantic environments *while aligning vector representations*?
- **A**: Well, sort of.
    - To avoid the subtask of aligning vectors, we could try to use (some form of) MT techniques
        - specifically, word alignment
    - Virtually all the statistical word alignment algorithms take context into some consideration
    - Although most of them do not look at syntactic relations
        - Chiang (2010) does
    - Some try to align at the phrasal level
- **All these methods require us to train each language pair individually**
    - Which might imply nontrivial effort — we need lots of data to get reliable results!

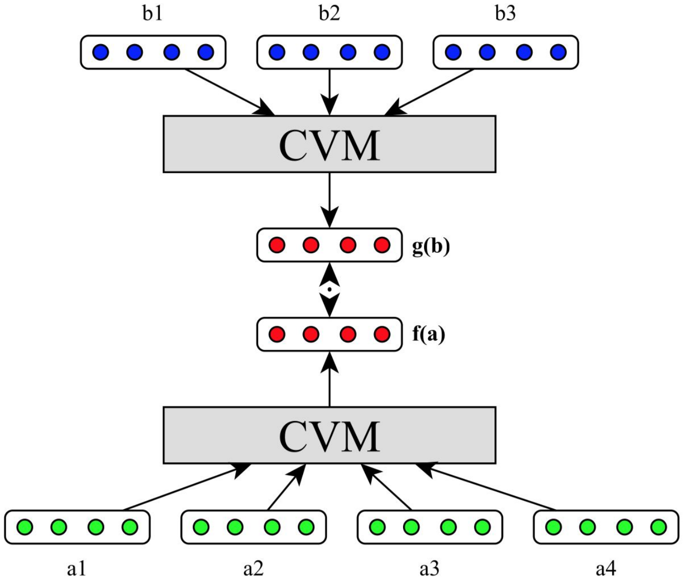
The Current State: the General Strategy — Solution?

- **Q**: Can we *try* to include syntactic and semantic environments
  - *While aligning vector representations* ?
  - Without training for each pair of languages?
- **A**: Well, yes!
  - Joint-space word embeddings! (Hermann and Blunsom 2014)
    - Vector representation
    - In a shared space *for all languages*
    - Trained on parallel text
    - Captures a semantic representation of the shared meaning
    - The composition functions (*f* and *g*) can include syntactic information



The Current State: the General Strategy — Solution?

- Joint-space embeddings will allow us
    - To train once with all the languages we need
        - We need a parallel corpus containing the ones we need (the paper uses the TED talks)
    - To implement **BRM**™ automatically in a relatively accurate way
    - To test properties of vector semantic spaces
        - Do FrameNet Frames partition the “semantic space” in a different way than the algorithm itself?
        - That is: **are projections of a Frame’s LUs “close together” or not?**



Practicalities: Graph Alignment and Restructuring

Going back to the graph alignment problem:

- What history has taught us:
  - xFNs branching off at different times
  - Parallel creation of frames in BFN and other xFNs
  - Adaptation of BNF Frames to similar Frames in other xFNs
  - “Backporting” of Frames from xFNs to BFN

Practicalities: Graph Alignment and Restructuring

How to learn from history (and ease the pain)

- We need a tool able to do the kind of restructuring previously outlined
  - Based on *formal methods* (se we can trust it 100%)
- Also: wouldn’t a *versioned database* be nice?
  - Able to to go back to any tagged version
    - And anything in between
  - Like Git, Mercurial, Svn, CVS, Darcs, ...
  - But **able to deal with FN elements** (Frames, FEs, LUs, Annotations) **at that level**
- And how about Constructions?
  - And the further restructuring that’s going to be needed?

Practicalities: Maintenance and Growth

- Q: How do we go from a *snapshot* alignment to *continuously aligned* MLFN?
    - That is, how do we manage the growth in time of the new resource?
    - More practically: who’s doing what?
        - Is ICSI supposed to deliver an infrastructure that the other projects can exploit?
        - Or should some data exchange format be defined?
        - Or anything in between?
- I hope we can discuss these issues — and others — later today
    - Feedback very much appreciated!
- Thank you!