Subsensing in dictionaries

A report for OASIS LEXIDMA
Author: Michal Měchura michmech@lexiconista.com October 2020

This report reviews the practice of subsensing (= embedding senses inside other senses) in dictionary encoding schemas, and makes a proposal for representing it in the LEXIDMA standard.

Introduction

One thing we often see in dictionary schemas is that they allow some form of recursive embedding, in other words, containing objects of one type inside objects of the same type. Typical examples include subsensing (a sense contains other, more specialized senses) and subentry grouping (for example the entry for walk noun and walk verb are subentries of another, larger entry).

This report deals specifically with subsensing but it is hoped that the observations will inform our thinking on all kinds of embedding.

Listing 1 shows an invented example of an entry with subsensing. Notice how some of the sense elements contain other sense elements.

Listing 1: A dictionary entry with subsensing

```plaintext
- entry
  - headword: bank
  - sense
    - definition: a place or institution where you can store something
    - sense
      - definition: an institution that stores money and provides financial services
        example: He got a large loan from the bank.
      - sense
        - definition: a supply of money in a game
          domain: gambling
        - sense
          - definition: collected storage of something valuable
            example: blood bank
      - sense
        - definition: a raised, elongated area of ground
          - sense
            - definition: the side of a river or canal
              example: I swam to the opposite bank.
          - sense
            - definition: a raised area of ground that slopes at the sides
              example: a bank of earth between two fields
```

How dictionary schemas enable subsensing

Subsensing by recursion is when, in our dictionary schema, we have a type such as sense and the schema allows instances of sense to contain other instances of sense. This phenomenon (when an object of one type is allowed to contain other objects of the same type) is called recursion. Recursion can go on into infinity, although in practice dictionaries rarely go beyond two levels.

Subsensing by subtyping is when, in our dictionary schema, we have two types such as sense and subsense and the schema says that instances of sense can contain instances of subsense. This isn't recursion, formally speaking, because the two types sense and subsense are not identical (and so it cannot go on into infinity). But it is similar to recursion because, in a typical dictionary schema that allows it, the two types sense and subsense tend to contain the same or similar types of content: definitions, translations, examples sentences and so on. We could say that the type subsense is a subtype of the type sense.

A special case of subsensing is something I would call supersensing. It is when a dictionary entry contains an otherwise flat list of senses, but the senses are grouped into "supersenses" according to some semantic or syntactic characteristic. What's typical for supertyping is that the supersenses tend not to contain the same types of content as senses (definitions, examples): they tend to be just brief signposts to help the user navigate through a long entry. Later in this document we will see examples of dictionaries that do this (it's Example 3 and Example 4, if you want to jump ahead).

What all these practices have in common is that they produce dictionary entries in which the senses are not a flat list but a hierarchy.

Is subsensing bad?

Subsensing, regardless of whether it is by recursion or by subtyping, has the effect that the same kind of information (definitions, example sentences etc.) ends up being located at different depths inside the entry (starting from whatever the top-level element is). This is a distracting complication for various digital agents (= software tools that process dictionary entries) such as dictionary writing systems or programs that extract data from dictionaries. From the perspective of someone who writes software tools for processing dictionary entries, it would be more convenient if each entry had only a flat list of senses with no embedding.

I would argue that this concern is non-trivial and that it should inform LEXIDMA's decisions, given that our mission is to make lexicographic data more easily processable by machines.

What is subsensing used for?
Why do dictionary schemas allow subsensing? Broadly, I see two kinds of motivation.

For modelling sense relations: specifically, the is-a or is-a-specialization-of relation. You have one broad sense (let’s call it the mother sense) and then you have a number of more detailed senses (let’s call them the child senses) which, metaphorically speaking, cover smaller stretches of the same territory. A popular choice with lexicographers to model this fact is by containing the child senses inside the mother sense.

For navigating large entries: when an entry has so many senses that skimming through all of them at once is too much for the end user, the lexicographer may decide to group them into a smaller number of ‘supersenses’ according to some (semantic, syntactic, ...) characteristic they have in common. The idea is that this will help the end user to navigate more quickly to the particular sense they have in mind.

These two motivations are driven by genuine requirements. When a dictionary entry has multiple senses, it is natural that there are some relations between them and that we want to communicate that fact to the end user. The only question is whether subsensing is the best or only way to model that. Similarly, when a dictionary entry has a large number of senses, it is uncontroversially a good idea to give the end user some way of navigating through the senses quickly: the only question is whether a hierarchical list of senses and subsenses is the best way, and if it is, whether the hierarchy needs to be hard-coded in the entry structure or whether it can be constructed at viewtime, as a presentation feature.

The proposal

Let’s say our goal is to find a way that we can satisfy the two requirements (telling the end user about sense relations and empowering the user to easily navigate long entries) but with only a flat list of senses, without embedding.

The solution is obvious: we keep the list of entries flat and we record the is-a-subsense-of relation in a stand-off fashion, like in Listing 2 below. For presentation (= for displaying the entry to an end-user) we can use the stand-off relations to dynamically reconstruct a tree-structure with senses and subsenses. For all other processing, we have a flat list of senses with no embedding.

Listing 2: A dictionary entry where subsensing is treated as sense-to-sense relations

```
- entry
  headword: bank
  - sense
    id: bank_1
    definition: a place or institution where you can store something
  - sense
    id: bank_2
    is-subsense-of: bank_1
    definition: an institution that stores money and provides financial services
    example: He got a large loan from the bank.
  - sense
    id: bank_3
    is-subsense-of: bank_1
    definition: a supply of money in a game
    domain: gambling
  - sense
    id: bank_4
    is-subsense-of: bank_1
    definition: collected storage of something valuable
    example: blood bank
  - sense
    id: bank_5
    definition: a raised, elongated area of ground
  - sense
    id: bank_6
    is-subsense-of: bank_5
    definition: the side of a river or canal
    example: I swam to the opposite bank.
  - sense
    id: bank_7
    is-subsense-of: bank_5
    definition: a raised area of ground that slopes at the sides
    example: a bank of earth between two fields
```

The senses and subsenses are encoded as a flat list, while their hierarchical arrangement needs to be inferred by following the chains of references between IDs (bank_1 etc). This makes the entry less legible for humans but more easily processable for machines.

The fact that the entry is now less legible for humans should not be taken as a disadvantage. This way of encoding entries (as a flat list of senses) is not intended for presentation to humans (neither to the end users nor to the dictionary editors). Each dictionary project can, if it wants to, keep their data internally as they already do, with embedded subsenses, like in Listing 1. The flat format suggested here is intended only for data interchange:

- To export entries from your internal format into the LEXIDMA standard, you need to flatten your sense hierarchy, give each sense an ID, and indicate the sense relations with ID-to-ID links.
- To import entries from the LEXIDMA standard, you optionally can, if you want to, reconstruct the sense hierarchy from the ID-to-ID links.

If the LEXIDMA standard becomes widely accepted, it is possible that some software systems will start using it as their internal native format for entry encoding. Even then, the ‘flat’ arrangement of the sense does not need to be what the end user sees or that the lexicographer works with.
To show the entry to end users, the software tool can first reconstruct the sense hierarchy from the ID-to-ID links (producing something like Listing 1) and then pretty-print that hierarchy for display to the user.

To present an entry to a lexicographer for editing, the software tool can first reconstruct the sense hierarchy from the ID-to-ID links (producing something like Listing 1) and make it available for editing. When the lexicographer is finished with the entry, the tool can flatten the list of senses, give each sense an ID, indicate the sense-to-sense relations with ID-to-ID links, and store this flattened version.

To summarize and repeat, the flattened version would be an internal representation for machine processing, while the ‘un-flattened’ version, with its hierarchy of senses and subsenses, would be for presentation to humans. In other words (and this terminology will be familiar to software engineers), the ‘un-flattened’ version as in Listing 1 is the view model while the flattened version as in Listing 2 is the domain model.

And now, let’s look at a couple of examples from real-world, born-digital dictionaries. In each example we’ll see how an entry’s sense hierarchy can be flattened and then ‘un-flattened’ without loss of information.

Example 1: sicher in DWDS

This entry comes from a digitised version of Wörterbuch der deutschen Gegenwartssprache. Some of the larger entries in this dictionary have a very “branchy” hierarchy of senses and subsenses which goes down to more than two levels, and sicher is one of them. Only the first few senses from ‘sicher’ are shown here.

Original (adapted for brevity and clarity)

```
- entry
  headword: sicher
  pos: adj
  - sense
    definition: nicht von Gefahr bedroht, ungefährdet
    example: ein sicherer Weg
    - sense
      pattern: vor etw|jmdm sicher sein
      example: hier seid ihr vor der Entdeckung sicher
      - sense
        expression: sicher ist sicher!
        definition: lieber vorsichtig sein, lieber nichts riskieren!
        example: ich nehme den Regenschirm mit, sicher ist sicher!
      - sense
        expression: Nummer Sicher
        definition: Gefängnis
        example: in Nummer Sicher sitzen
    - sense
      definition: zuverlässig, verlässlich
      example: ...
```

If for the moment we ignore the fact that some of the subsenses are in fact phraseological multi-word entities and that we would probably prefer such entities to be encoded as separate entries, then this sense hierarchy could be flattened as follows.

Flattened
Example 2: **work** in Lexico.com

Note: Lexico.com is where Oxford University Press now publishes its online monolingual English dictionary for native speakers (as opposed to its monolingual English dictionary for learners, for that see the next example).

This is a composite entry with a subentry for ‘work’ noun and a subentry for ‘work’ verb — I assume these would be treated as separate entries in the LEXIDMA standard, so here we see only the noun entry. The same goes for the multi-word units at the end of the web page: I assume these would be treated as separate entries in the LEXIDMA standard.

The entry for ‘work’ noun has two levels of senses and subsenses. Only the first few senses are shown here.

Original

- entry
  headword: work
  pos: noun
  - sense
    definition: Activity involving mental or physical effort done in order to achieve a purpose or result.
    example: he was tired after a day’s work
    - sense
      definition: Mental or physical activity as a means of earning income; employment.
      example: I’m still looking for work
    - sense
      definition: The place where one is employed.
      example: I was returning home from work on a packed subway
    - sense
      definition: The period of time one spends in paid employment.
      example: he was going to the theatre after work
    - sense
      definition: A job.
      label: West Indian
      label: count noun
      example: I decided to get a work
    - sense
      definition: A task or tasks to be undertaken.
      ...
- entry
  - headword: work
  - pos: noun
    - sense
      - id: work_noun_1
        definition: Activity involving mental or physical effort done in order to achieve a purpose or result.
        example: he was tired after a day’s work
    - sense
      - id: work_noun_2
        definition: Mental or physical activity as a means of earning income; employment.
        example: I’m still looking for work
    - sense
      - id: work_noun_3
        definition: The place where one is employed.
        example: I was returning home from work on a packed subway
    - sense
      - id: work_noun_4
        definition: The period of time one spends in paid employment.
        example: he was going to the theatre after work
    - sense
      - id: work_noun_5
        definition: A job.
        label: West Indian
        label: count noun
        example: I decided to get a work
    - sense
      - id: work_noun_6
        definition: A task or tasks to be undertaken.

Example 3: **work** in Oxford Advanced Learner’s Dictionary

This is the entry for ‘work’ verb (there is a separate entry for ‘work’ noun on another web page). This is actually an example of supersensing (which, as you will recall, I have defined as a special case of subsensing) because there is a flat list of senses, numbered from 1 to 13, but the senses are grouped into what we could “supersenses”, each introduced by a brief signpost: senses 1 and 2 are under the supersense “do job/task”, senses 3 and 4 are under the supersense “make effort”, and so on. Again, only the first few senses are shown here.

Original (I assume that this is how, in broad strokes, the entry is encoded internally)
- entry
  headword: work
  pos: verb
  - supersense
    signpost: do job/task
    - sense
      definition: to do something that involves physical or mental effort, especially as part of a job
      example: I can't work if I'm cold.
    - sense
      definition: to have a job
      example: Both my parents work.
  - supersense
    signpost: make effort
    - sense
      definition: to make efforts to achieve something
      example: She dedicated her life to working for peace.
    - sense
      pattern: work yourself/somebody + adv./prep.
      definition: to make yourself/somebody work, especially very hard
      example: She works herself too hard.
  - supersense
    signpost: machine/device/system
    ...
speech. Unlike the previous example, the supersenses contain no definitions or signposts or indeed any content at all, apart from the part-of-speech label followed by the senses. The part-of-speech label is then repeated in each sense, so its presence in the supersense is completely redundant: it serves only as a label for grouping senses together, in order to empower the user to navigate a large entry more easily.

So this, even though it appears to be an example of subsensing, really isn't one. When exporting this dictionary into the LEXIDMA standard, we should be probably export each part-speech-block as a separate entry, like we did in Example 2.

Entries in this dictionary also contain multi-word subentries. As in previous examples, these should probably be treated as separate entries in the LEXIDMA standard.