

Relations

Some core semantic and ontological relations between entities are highlighted in the CASTEMO data model.

- [Relations overview](#)
- [Superclass \(SCL\)](#)
- [Superordinate Entity \(SOE\)](#)
- [Classification \(CLA\)](#)
- [Identification \(IDE\)](#)
- [Synonym \(SYN\)](#)
- [Antonym \(ANT\)](#)
- [Holonym \(HOL\)](#)
- [Property Reciprocal \(PRR\)](#)
- [Action/Event Equivalent \(AEE\)](#)
- [Implication](#)
- [Subject/Actant1 Reciprocal \(SAR\)](#)
- [Actant semantics: Subject Semantics \(SUS\), Actant 1 Semantics \(A1S\), and Actant 2 Semantics \(A2S\)](#)
- [Related \(REL\)](#)

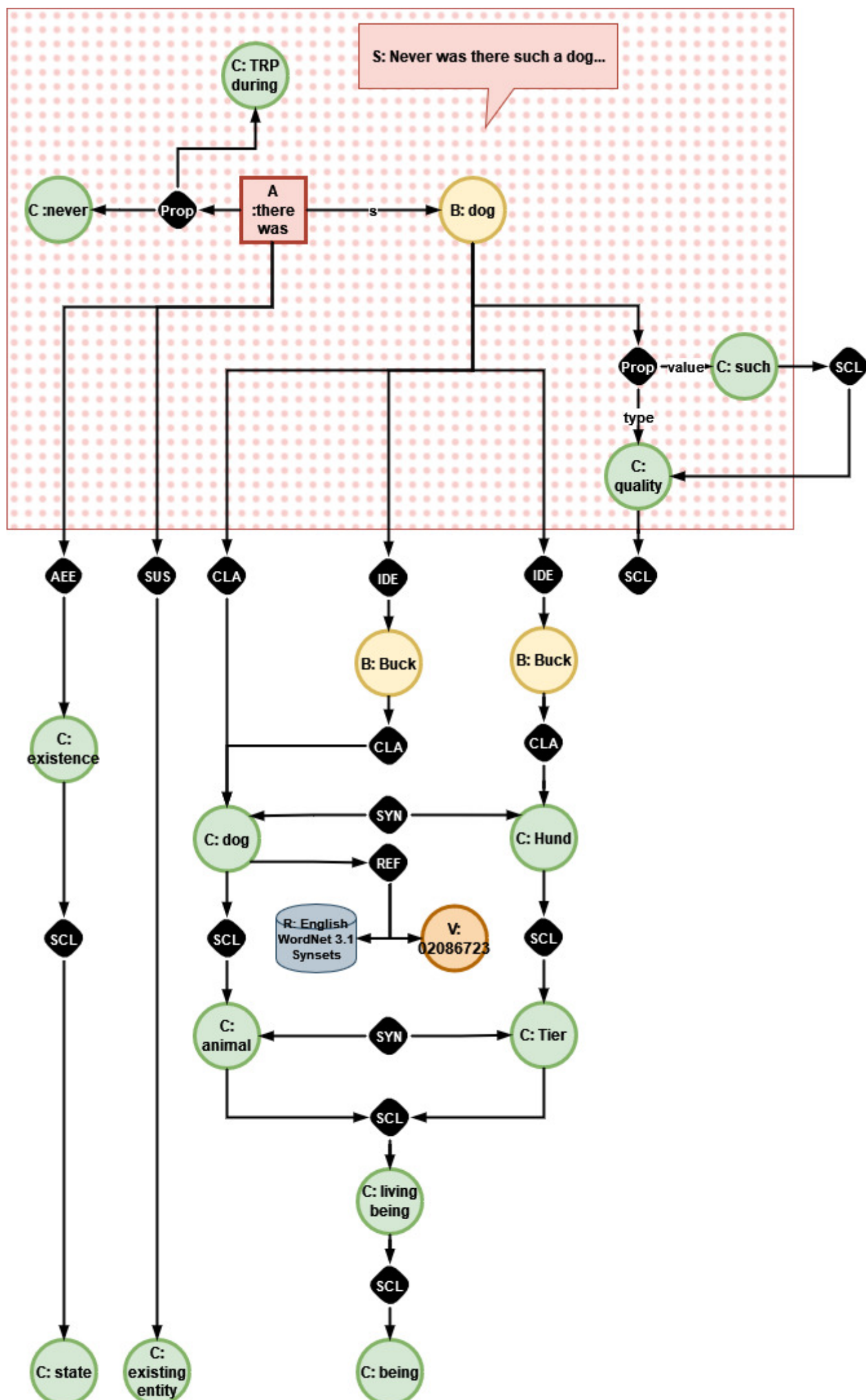
Relations overview

Some core semantic and ontological relations between entities are highlighted in the CASTEMO data model. Their reserved nature - in ways unavailable for specific Property Types, which depend on specific UUIDs - ensures their compatibility across CASTEMO research databases (and InkVisitor deploys). The following table gives an overview of all the relations, the entity types which allow that relation, a brief definition, and an example.

Relation name	Abbreviation	Name in JSON code	Inverse relation name	Allowed entity combinations	Detail	Example
Superclass	SCL	Superclass	Subclasses	A-A, C-C	Superordinate term (hypernym).	C apple -> C fruit A walk -> A move
Superordinate Entity	SOE	Superordinate Entity	Subordinate Entities	L-L, O-O, G-G, E-E, S-S, E-S, S-E, R-R (under discussion: T-T)	Ontological superset.	L Milan -> L Italy O the handle of this door - O this door
Synonym	SYN	Synonym	Synonym	A-A, C-C	Synonym both within a language and across languages (equivalent).	C funny - C strange
Antonym	ANT	Antonym	Antonym	A-A, C-C	Opposite term.	C good - C bad
Property Reciprocal	PRR	PropertyReciprocal	PropertyReciprocal	C-C	The concept that the property reciprocates if read the other way.	C parent <-> C child

Subject/Actant1 Reciprocal	SAR	SubjectActant1Reciprocal	SubjectActant1Reciprocal	A-A	The action that the actant1 gives back to subject.	A hear (from sb - about st) <-> A tell (sb - about st)
Subject Semantics	SUS	SubjectSemantics	Used as Subject semantics	A-C	Semantics of the subject (actant 0) slot.	A talk (to sb - about st) -> C speaker
Actant1 Semantics	A1S	Actant1Semantics	Used as Actant 1 Semantics	A-C	Semantics of the actant 1 (object 1) slot.	A talk (to sb - about st) -> C listener
Actant2 Semantics	A2S	Actant2Semantics	Used as Actant 2 semantics	A-C	Semantics of the actant 2 (object 2) slot.	A talk (to sb - about st) -> C speech content
Action/Event Equivalent	AEE	ActionEventEquivalent	Action equivalent	A-C	What is this action in the world of nouns?	A baptize - C baptism
Related	REL	Related	N/A	SPECTRAL OGB- SPECTRAL OGB	Free relation to connect things not connected by more specific semantic relations.	C tea -> C afternoon
Classification	CLA	Classification	Instances	PLOGESTR B-C	Relations of specific entities to the concepts they are the instance of.	O this apple -> C apple

Identificati on	IDE	Identification	Identification	PLOGESTR B- PLOGESTR B	Declaration of ontological identity between things.	L Italie (French) <- > L Italy (English)
Holonym	HOL	Holonym	Meronyms	C-C	Relation of a part to its whole.	C gate of a monastery -> C monastery
Implication	IMP	Implication	Used as Implication	A-A	Action implied by this action.	A dine (with sb) - > A be in the company (of sb)
In- statement Classificati on			In-statement instances	PLOGESTR B-C	Same as Classificati on, but done from within a Statement.	
In- statement Identificati on			In-statement Identification	PLOGESTR B- PLOGESTR B	Same as Identificati on, but done from within a Statement.	



Superclass (SCL)

Superclass (SCL) is a semantic relation which relates an Action to one or more Actions, or a Concept to one or more Concepts. It denotes **hypernym (genus proximum)**, i.e. the more generic conceptual class to which an Action or Concept belongs.

Example: `C apple - SCL - C fruit`.

Do not confuse Superclass with Classification. Classification serves to relate specific entities (PLOGESTRB) to its Class (C). All of these specific entities are then Instances of this Class.

Superordinate Entity (SOE)

Superordinate Entity (SOE) is a Relation which connects a subordinate entity to an entity in which it is fully contained.

Example: `L Milan - SOL - L Italy`.

In CASTEMO, only the direction from the subordinate to the superordinate entity is being stored. However, subordinate entities are displayed in InkVisitor to give an overview of the data.

Classification (CLA)

Classification (CLA) is a Relation between a specific PLOGESTRB entity and the class (Concept) to which it belongs.

Example: 0 this apple - CLA - C apple; 0 another apple - CLA - C apple.

Classification is **vital for querying the database**: it allows to find specific instances of a generic Concept (e.g., all apples) but also, through the Superclass tree of Concepts, more general classes (e.g., any instances of fruit).

Do not confuse Classification with Superclass.

Identification (IDE)

Identification (IDE) serves to declare the **identity** between PLOGESTRB entities, both **within an entity type** (e.g. `P Rocket Man - IDE - P Elton John`) and **across entity types** (`L the boat on which we got married - IDE - 0 this same boat that we then sold`).

Not unlike Synonymy, it helps preserving the original expressions used in the modelled Resources, while still keeping them identical. However, Synonymy pertains to Actions and Concepts and is a *semantic* relation, while Identification pertains to entities of this world, and is an *ontological* relation.

Unlike all other Relations, Identification can be done with a **Certainty level**, i.e. the identification can be done at a lower Certainty level, e.g. "probable", "possible", or even "false", which is a negative declaration that the two entities are *certainly not* identical, which can sometimes be useful if some resource has made this (erroneous) identification.

Any queries of entities must take into account the Identifications of the entity (usually including the definition of threshold certainty levels), if they want to include also all of its identicates in search results.

Entities with which an entity is identical are called **identicates**.

Synonym (SYN)

The CASTEMO data model recommends a **strong understanding of synonymy**. For two [lexemes](#) to be related with the Synonym Relation, they need to have the **same Superclass** (if there are more superclasses, they need to share all of them). Also, if they are related to a meaning bank, **the meaning ID in the relevant meaning bank needs to be the same**.

This strong understanding allows to consider synonymy as **transitive**: if A is synonym of B and B is synonym of C, then A is synonym of C. (In semantic network terms, this would be described as *triadic closure*.)

The Synonym Relation is also used for **cross-language equivalence**, e.g. `C dog (English) - SYN`
`- C Hund (German)`.

The Synonym relation helps to **extend database queries** and also include the Synonyms of Concepts and Actions queried. Through Concepts used in Classification, synonymy can also help build **sets of entities for analysis** (e.g. including all Living Beings classified either as "dog (English)" or as "hound (English)" or as "Hund (German)").

Antonym (ANT)

The CASTEMO data model recommends a **strong understanding of antonymy**, i.e. one which to some degree applies also to the Superclasses of both lexemes.

Through **Concepts used for Classification**, antonymy helps to build **contrasting sets of entities for analysis**. Through **antonymous Actions**, it helps to build contrasting sets of Events.

Holonym (HOL)

Holonym (HOL) Relation denotes the relation between a Concept representing a part of something to a Concept representing a whole.

Example: `C door of a house - HOL - C house`.

The reverse relation is called meronymy.

In CASTEMO knowledge graphs, we only store the relation from the part to the whole, not the other way around, but the reverse direction is displayed in InkVisitor.

This ontological relation between Concepts is analogical to what the SOE Relation is between PLOGESTRB entities.

Property Reciprocal (PRR)

Property Reciprocal (PRR) is a Relation connecting two Concepts which can feature as a Property Type. In a Property, composed of Origin, Property Type, and Property Value, it defines what Property Type the Property Value would give back to Origin if the Property would be stored the other way. For instance, the PRR of C "mother" will be "child", because if `P Bernard - has - C mother - P Anna`, then `P Anna - has - C child - P Bernard`. Similarly, the PRR of C "occupation" will be "occupation holder", because if `P Bernard - has - C occupation - C baker`, then `C baker - has - C occupation holder - P Bernard`.

Property Reciprocal serves to symmetrize the Properties in a knowledge graph derived from the Collected Data Database by enriching them with the Property sent the other way, thus enabling to **query CASTEMO data more easily** and avoid missing properties which happen to be encoded the other way around.

Action/Event Equivalent (AEE)

The **Action/Event Equivalent Relation (AEE)** connects always one Action to one Concept, and it serves to translate between the world of verbs and the world of nouns. For instance, A "to baptize" would have the AEE C "baptism".

Since any specific Event which happens to be a baptism should have the Classification C "baptism", all specific "baptism" Events and all Statements with the Action "to baptize" used in the action slot can be **queried together** in the database, as long as they have the AEE and CLA properly set.

Just as importantly, AEE is **a way for verbs to acquire coherent Superclasses**. While the Superclass Relation is allowed for Actions, most verbs do not have manifest Superclasses. For instance, A "to baptize" does not have any Superclass such as *"to ritualize", while C "baptism" can acquire quite a straightforward Superclass tree (e.g. C "Christian ritual", which in turn will have the Superclass C "ritual"). These Superclasses propagated to Actions from Concepts through AEE are shown under Actions in InkVisitor, thereby helping coders to decide whether an Action fits the specific context. They are of course vital for **analytical queries** which aim at retrieving, for instance, all rituals, be they expressed through verbs (i.e., through an Action) or nouns (i.e., as Events).

Implication

Implication (IMP) is a Relation which connects an Action to one or more other Actions. It denotes an action which, by implication, must have happened because it is logically implied by the original action. For instance, `A travelled (with sb) - IMP - A was in the company (of sb)`.

Implication allows to **enrich data** in the knowledge graph with actions which, while not explicitly denoted by the verbs used, happened by implication. This in turn helps to extend the query results in cases where the Action's Superclass or Action/Event Equivalent did not necessarily cover the implied action.

Depending on the strictness of use of IMP and the research purpose, IMP can be used to produce only the first-level implication directly appended to the Action, or to follow the implication chain even further (e.g. if action A implies action B, and action B implies action C, then we may decide to also make action A imply action C upon data projection).

In linguistics, this relation type is sometimes also called **entailment**.

Subject/Actant1 Reciprocal (SAR)

The **Subject/Actant1 Reciprocal (SAR) Relation** relates two Actions. It is a type of Implication, but one which goes in the reverse direction: from actant 1 to the subject. For instance, if P Peter - A accompanied - P Susan, then P Susan - A was accompanied - P Peter.

Some SAR are **symmetrical**, i.e. they imply the reciprocation of the same Action from actant 1 to the subject (e.g., "was in the company (of sb)"). Some are **asymmetrical**, i.e. the reciprocated Action is different.

Actant semantics: Subject Semantics (SUS), Actant 1 Semantics (A1S), and Actant 2 Semantics (A2S)

Subject Semantics (SUS), **Actant 1 Semantics (A1S)**, and **Actant 2 Semantics (A2S)** are Relations each of which connects an Action to one or more Concepts **describing semantically the role of the holder of the actant slot**. E.g., A "to speak (to sb - about st/sb)" would have the SUS: C "speaker", A1S: C "participant in a conversation", and A2S: C "speech content".

These Relations describing actant semantics then allow to **study the roles of entities holding that actant slot**. E.g., we can inspect all women who have either the in-statement CLA C "preacher", or who are known to have A "preached", because A "preach" would have exactly the SUS: C "preacher". Therefore, the actant semantics enrich actant role holders with the respective roles implied by the Action.

These relations also allow to properly understand the **direction of action**: e.g., a text can say that `P Lucy - A listened - (to) P Mary`, or that `P Mary - A spoke - (to) P Lucy`. The relevant actant semantics will allow to identify, in both wordings, Mary as the speaker and Lucy as the listener. Thus, for research purposes, we can for instance safely produce a directed conversation tie going from Mary to Lucy rather than the other way round from any of the two wordings, and also harbour the **passive and active voice** correctly under the same roof.

Actant semantics are the CASTEMO way of ensuring **semantic role labelling**: it is done by properly defining any Action type's semantic valency, rather than losing time with labelling semantic roles in specific Statements.

Related (REL)

Related (REL) is the least specific Relation which allows to relate entities of any type by way of association.

Example: `C tea - REL - C afternoon`

For more specific semantic relations, a relevant specific Relation or Property should be preferred.