

# The DQD Query Language

LCP's query language DQD (Descriptive Query Definition) follows the idea of Entity-Relationship models. Entities sets are defined by logical formulae on properties and relations between them. The query engine then searches for those constellations inside the corpus or corpora selected. Along the lines of first order logic, quantors can be employed to enforce the existence or non-existence of constellations.

Every query needs to specify at least one result set, which is either a (plain) list of entities comprising the query matches, a statistical or a collocational analysis.

## Introduction

In a text corpus with standard annotations (viz the [CoNLL-U format](#)), a simple query for the occurrence of the word "dogs" would look like this:

```
Segment s
Token t
  form = "dogs"
Result => plain
  context
    s
  entities
    t
```

## Quick Reference

### Entities

Entities are defined in the [Corpus Template](#). Though aleatory names can be used, we use *Document* for documents, *Segment* for sentence segments and *Token* for tokens in the examples. Entities are expected to start with an uppercase character.

### Attributes

Attributes are also defined in the [Corpus Template](#) and thus their naming is free for a corpus creator to define. As a standard set, we use *form* for word forms, *lemma* for lemmas, *upos* for [Universal part-of-speech tags](#), and *morph* for [Universal features](#). Attributes stored as [Meta Data](#) is mapped to entity attributes unless a native attribute with the same name exists. In that case, the meta attribute needs

to be explicitly referenced.

From:

<https://liri.linguistik.uzh.ch/wiki/> - **LiRI Wiki**

Permanent link:

<https://liri.linguistik.uzh.ch/wiki/langtech/lcp/dqd/start?rev=1709644532>

Last update: **2024/03/05 13:15**

