Extending the Number Names – Naturally

Robert Superty
London, UK
superty@post.harvard.edu

Part 0: Introduction

There's a long tradition of recreational linguists and mathematicians devising new ways to name very large numbers in the English language, and many of these attempts have been published in Word Ways [1], [2], [3], [4], [5]. These names, however clever or well thought out, tend to end up as curiosities, as most very large numbers are generally only needed in technical or scientific contexts, where they are expressed as exponents rather than spelled out as words. A chemistry text that defines Avogadro's constant as "approximately six hundred two sextillion moles^{-1}" (rather than the much more concise 6.02 x 10^{23} mol^{-1}) would be a rare find.

Yet there is something that seems to attract hobbyists again and again to invent new words for large numbers. Perhaps for the more linguistically inclined it's the exoticism of words like "quattuordecillion," or the feeling that a gap in the vocabulary is waiting to be filled, or, for the more numeric, the mind-boggling quantities that these words represent, or the challenge of building the most logical or complete system possible. For some, hearing the *name* of a large quantity helps to make it more real somehow.

Some of these extended names have seen some limited use in a variety of places. A whole genre of smart phone "clicker games" have become popular, one called *AdVenture Capitalist* comes to mind, where the goal is to earn absurd quantities of cash to become a "trigintillionaire" or a "quinquagintillionaire" [6]. (Usually this is done by repeatedly tapping on the screen – if only it were that easy.) It's not unheard of for a magazine article to throw in a very large number name to reinforce or exaggerate a large quantity being reported. In an article about a new Rubik's Cube record, for example, Gizmodo pointed out that there are over "282 trevigintillion (10^72)

permutations!" [7]. As the world becomes more connected in this information age, and volumes of data grow exponentially, maybe these large number names have finally found their day in the sun, and it may be just the right time to standardize them.

Here I will continue in the number nomenclature pursuit and propose a new system of large number names. But rather than just presenting a long list of names for powers of ten, or one thousand, or one million, there's a bit of explanation to do. The first part of this article will aim to remind the reader of the current "official" English names for numbers that are defined in most unabridged dictionaries. This covers words like "vigintillion" and "centillion" that may or may not already be familiar. The second part will then consider how these existing names might be extended, establishing some general principles. Next, I will consider some already proposed solutions to extending the number system, and finally, based on these principles, I will present a new system of names taking a different approach from the existing systems, while at the same time being almost fully compatible with most.

Part I: The Dictionary Number Names – A Refresher

When most people think of very large number words, the familiar "-illion" words will come to mind—words like "million," "billion," and "trillion," and maybe a few less common ones like "quadrillion" for the especially keen. These "-illions" represent the start of a sequence of twenty-one such words that are defined in many unabridged English dictionaries, all based on the original word "million." The sequence of "-illion" words form the backbone of any large number name, and so we must focus on these words in any large number naming system.

"Million" itself has come into English from Latin, via Italian and French [8]. The "milli-" part comes from Latin *mille*, meaning "one thousand." The "-ion" comes from an augmentative suffix, so the parts together mean "large thousand."

Subsequent "-illion" words are actually portmanteaux (think "bridezilla"!) of Latin number prefixes (again, generally via French) and "million," with the "m" replaced by the relevant number prefix.

The first ten are as follows:

Index	Prefix	"-illion" Word	Index	Prefix	"-illion" Word
1	-	million	6	sexti-	sextillion
2	bi-	billion	7	septi-	septillion
3	tri-	trillion	8	octi-	octillion
4	quadri-	quadrillion	9	noni-	nonillion
5	quinti-	quintillion	10	deci-	decillion

These prefixes should be fairly recognizable to most English speakers, even those with little knowledge of Latin, at least up to the tenth or so "-illion," as they are used in many words, for example "bicycle," "triple," "quadrilateral," "quintuple," "sextet," "September," "octopus," "nonagon," and "decimal."

We have not yet said anything about the actual values of these numbers. This is a point of contention, although the question is fairly settled in the English language at least. Originally, the names were intended to be powers of one million, meaning that since one million represented 10^6, or one with six zeroes, one billion represented 10^{12}, trillion represented 10^{18}, and so on. Any given number name was one million to the power of its index, and by extension, ten to the power of six times the index, or 10^{6N}, with "N" representing the index or root word. This is the system used in most non-English speaking countries today and is called the "long scale."

However due to a historical quirk in English-speaking countries these values have almost universally come to represent sequential powers of one thousand instead of powers of one million (meaning a new "-illion" word every three digits rather than six). This leads to an offset between the thousand power and the root word: million is now the second power, billion, the third, and so on. The value of each number than comes to be ten to the quantity of three times the index *plus three* (the offset), or 10^{3N+3}. This is unfortunate, and

undermines the natural connection between the root and the numeric value, but it is the common definition of these words today.

This article will focus on the index and sequence of these words, and will leave it to the reader to convert to the short scale or long scale value as they see fit. The systems discussed will work in both versions; here it is generating the words that matters. (On the rare occasions where I do refer to specific values, they will tend to be in the short scale unless otherwise noted.) Rather than awkwardly calling these words "-illions," from now on, when I refer to them collectively, I will co-opt the fictitious number word "zillion" and call these sequential number words "zillions." This is adopting the practice that John Conway and Richard Guy use in their seminal *Book of Numbers* [9].

Onward to the next ten zillion words. The zillions with indexes 11-20 are based on Latin cardinal numbers:

Index	Latin Number	Zillion	Index	Latin Number	Zillion
11	*undecim*	undecillion	16	*sedecim*	sexdecillion
12	*duodecim*	duodecillion	17	*septendecim*	septendecillion
13	*tredecim*	tredecillion	18	*duodeviginti*	octodecillion
14	*quattuordecim*	quattuordecillion	19	*undeviginti*	novemdecillion
15	*quindecim*	quindecillion	20	*viginti*	vigintillion

One can see that these names are not all exact matches, and are not perfect Latin, perhaps due to errors when they were first conceived and used, or perhaps they were originally based on non-standard variants of classical Latin. In reality we shouldn't expect them to be perfect Latin – they are now English words.

The last defined zillion is:

Index	Latin Number	Zillion
100	*centum*	centillion

Centillion is the highest defined number word that we have in the zillions system, but it should be mentioned that there are two additional large number words found in most dictionaries: "googol" (10^{100}) and "googolplex" (10^{googol}). I won't really touch on them in this article, as they are one-off names for single numbers, and cannot be combined with other number words to integrate into the schemes we will discussed, which are based on zillions. (Of course any good extension to these number names should at least be able to name the number represented by "googol," and even "googolplex" depending on the range of the system!) [10]

Part II: Principles for Completing and Extending the System

Let's pause to take stock of what we have to work with. We have zillion names for the first through twentieth zillions, and then, floating by itself, we have a defined word for the one hundredth zillion. These are based on prefixes for numbers like "bi-"and "tri-" that are ultimately derived from Latin, that worked their way into English via French, or, for the larger zillions, on Latin cardinal numbers more directly, albeit with some variation from the original classical forms.

These dictionary names allow us to name all consecutive numbers up to (in the short scale) $10^{66} - 1$, the name of which starts as "nine hundred ninety-nine vigintillion, nine hundred ninety-nine novemdecillion," and so on, descending through the zillions, and then after a very large gap, we can start again at one centillion and name some numbers higher (none higher than the 101^{st} zillion, however).

So when we talk about extending the number system, we not only mean naming zillions higher than one centillion, but also *completing* the system: filling in the gap between vigintillion and centillion. The goal here will be to examine options for (and ultimately create) a canonical system of zillions that is consistent with what is already present in the dictionaries, and which completes and extends this system in a logical and natural way.

Out of scope of this article are the more novel approaches to naming large numbers that are not based on the existing Latin system of zillions, systems which disregard what already exists and start over. Examples of note are Donald Knuth's "-yllions" system, where each successive term is the square of the previous [11], or Russ Rowlett's Greek-based system, which ditches Latin number roots in favor of Greek [12]. While these approaches can be well thought out and interesting to explore as alternative systems, they do not accomplish what we have set out here, to be harmonious with what is already in use.

In fact, I will set out six brief principles that any system must meet in order to be a candidate for our canonical zillions system:

1) The system must be backwards-compatible with defined English zillions
2) The system should be logical and self-consistent, with few one-off exceptions
3) There can be no gaps in the named numbers. If X has a name, all numbers less than X will also have names
4) Words must not be excessively long, or contain awkward letter combinations; basically these should feel like they could be plausible English words
5) As few new words should be defined as possible, and when a new word is created, it must be etymologically sound and strongly justified
6) The system should be (relatively) simple and natural, without very complex rules

Any system that meets principles 1 and 2, namely a self-consistent, backwards-compatible system, will be at least loosely Latin-based, given they will be a continuation of the current zillions. I will propose a new way to complete and extend the system of number names, and I will compare this effort to some other prominent proposals. Before we get into the unique aspects of this new "canonical" system, let's quick look at some zillion names that are common to all Latin-based systems.

Part III: The Standard Latin Zillions

It's time to list some new zillion words. But here, since these names are *not* present in any reputable dictionary, according to principle 5, we must make sure they are strongly justified and unambiguous. This is not a creativity exercise! The zillions that exist today are all based on Latin numbers that have single word names. Part of the reason the system stops at the twenty-first zillion is that the Latin number for twenty-one, *viginti unus*, consists of multiple words, and there is no obvious way to make this into a single zillion word. There are, however, larger numbers that *are* single words, and that are easy to turn into zillion words in a mechanical and standard way, namely: remove the last vowel and add "-illion."

First, zillions based on the Latin multiples of ten, which will replicate the exact pattern for "vigintillion":

Index	Latin Number	Zillion	Index	Latin Number	Zillion
20	*viginti*	**vigintillion**	60	*sexaginta*	sexagintillion
30	*triginta*	trigintillion	70	*septuaginta*	septuagintillion
40	*quadraginta*	quadragintillion	80	*octoginta*	octogintillion
50	*quinquaginta*	quinquagintillion	90	*nonaginta*	nonagintillion

Next, zillions based on the Latin multiples of one hundred, which will be based on the pattern for centillion:

Index	Latin Number	Zillion	Index	Latin Number	Zillion
100	*centum*	**centillion**	600	*sescenti*	sescentillion
200	*ducenti*	ducentillion	700	*septingenti*	septingentillion
300	*trecenti*	trecentillion	800	*octingenti*	octingentillion
400	*quadringenti*	quadringentillion	900	*nongenti*	nongentillion
500	*quingenti*	quingentillion			

It's important to re-emphasize that there is not much inventiveness going on here—there is a pretty direct pattern already established by vigintillion and centillion that allows us to extrapolate without really having to create new

forms, and the system is very conservative. In fact, it makes the dictionary zillions *more* consistent: why should vigintillion be a number name but not trigintillion, or centillion but not ducentillion?

Latin has one more single-word number name that can be made into a zillion:

Index	Latin Number	Zillion
1000	*mille*	millillion

Remember that the very first zillion, million, also has *mille* as its root, so we have in a sense come full circle here!

We now have a more fleshed out set of zillion words, from the first to the twentieth, and then round numbers up to one thousand. The zillion words we now have, the dictionary names plus these additional tens and hundreds and thousand, I will call the "Standard Latin Zillions." But there are still multiple gaps for the zillions in between. The first challenge is to fill those gaps, where we don't have a single-word Latin name to directly guide us, and then to extend the names beyond the thousandth zillion.

It is in these two tasks that the existing systems diverge, with no real standard. We will briefly look at two existing examples, and then in Part VI I will propose a new solution.

Part IV: The Strict Latin Systems of Henkle and Ondrejka

In the very first issue of *Word Ways*, Dmitri A. Borgmann listed out a series of large number names that were proposed by a "Professor Henkle" and published in a book called *The Philosophy of Arithmetic*, by Edward Brooks. [1] (A republishing of a republishing.) A call was put out to the readers to critique and improve the system, and in a 1968, a system proposed by Rudolf Ondrejka was published, which refined the Henkle list, ironed out some inconsistencies, and harmonized it with the dictionary number names. [2] We will discuss Ondrejka's variant and refer to this as the Ondrejka/Henkle system.

The hallmark of this system is that it pays much attention to the specifics of the Latin number prefixes being used. It uses the existing dictionary forms for the zillions from one to twenty, and the Standard Latin Zillions for the multiples of ten and one hundred (with one tiny variant: "sexcentillion" instead of "sescentillion"). In doing so, it notes that the majority of the zillions from one to nine are based on Latin *ordinal* forms, while the higher zillions are based on *cardinal* forms. Therefore, to define the zillions between, say, the twentieth and thirtieth, the system combines an ordinal form for the units, and a cardinal form for the tens. For example, the twenty-first zillion is listed as "primo-vigintillion," literally "first twenty-illion." The unit prefixes go on: "secundo-," "tertio-," "quarto-," "quinto-," "sexto-," "septimo," "octavo-," "nono-." These prefixes combined with the ten forms allows you to name all zillions up to centillion, closing that first gap.

The same principle is applied between the hundreds, but this time you use a units ordinal and a tens ordinal combined with the standard hundreds zillion. To take one example, we have, for the two hundred forty-ninth zillion: "nono-quadragesimo-ducentillion." The teens also have special irregular ordinal forms, presumptively based on Latin practice. (In reality these Latin ordinal forms seem to be far from standardized.)

To extend above "millillion," multiplier prefixes are introduced, creating a third tier of root words. An example from this range would be the 123,456th zillion:

sexto-quinquagesimo-quadringesimo-tri-vici-centi-millillion

This system is very appealing for its sense of logic and its faithfulness to the Latin origins of these zillion words. Up to one hundred, it is fairly simplistic and regular, requiring only ten additional prefixes in addition to the zillions defined up to that point. By the time one gets to the larger zillions, however, the system requires over one hundred unique prefixes, very few in regular use in English. The resulting combined forms sound very "Latin"—try reading the

long example above out loud. In fact, let's apply each of the principles to the system:

1) The system must be backwards-compatible with defined English zillions
    a. MET
2) The system should be logical and self-consistent, with few one-off exceptions
    a. MET – the system does have defined, logical rules
3) There can be no gaps in the named numbers. If X has a name, all numbers less than X will also have names.
    a. MET
4) Words must not be excessively long, or contain awkward letter combinations; basically these should feel like they could be plausible English words
    a. NOT MET – this is where the Ondrejka/Henkle system falls down. The prefixes are so faithful to Latin that they don't feel like English words. The forms are very long and contain multiple long hyphenated sections, which is not common in English words.
5) As few new words should be defined as possible, and when a new word is created, it must be etymologically sound and strongly justified
    a. PARTIALLY MET – the prefixes are closely tied to Latin and the etymology is well thought out, but regardless there are over one hundred new prefixes defined, which is not desirable.
6) The system should be (relatively) simple and natural, without very complex rules
    a. NOT MET – there are too many prefixes to memorize, and it's not immediately obvious when to use each one and in which order without a lookup table.

So it doesn't seem we can use the Ondrejka/Henkle approach to name our interim zillions or expand the system. Let's move on to the next example.

Part V: The Conway/Guy/Wechsler System

In their 1996 book, *The Book of Numbers*, renowned mathematicians John H. Conway and Richard Guy propose a number system that they co-devised with Allan Wechsler that allows the naming of any integer [9]. This system is probably the most authoritative to-date, and it has seen a decent amount of traction on the Internet.

In order to name the interim zillions, this system takes inspiration from the form of the teen zillions (undecillion, duodecillion, etc.) as a prefix + decillion, and extrapolates this to higher zillions. If you can have undecillion, why not "unvigintillion"? The system goes a step further, and uses phonetic rules to alter the spellings of these prefixes depending on the word they are being appended to, to ensure two things: 1) a fluid pronunciation, and 2) unique forms in all instances.

The Book of Numbers presents a chart, which I will reproduce here, showing how these prefixes are combined:

	units	tens	hundreds
0	-	-	-
1	un	[n] deci	[nx] centi
2	duo	[ms] viginti	[n] ducenti
3	tre [*]	[ns] triginta	[ns] trecenti
4	quattuor	[ns] quadraginta	[ns] quadringenti
5	quinqua	[ns] quinquaginta	[ns] quingenti
6	se [*]	[n] sexaginta	[n] sescenti
7	septe [*]	[n] septuaginta	[n] septingenti
8	octo	[mx] octoginta	[mx] octingenti
9	nove [*]	nonaginta	nongenti

Firstly, interim zillions are formed from the above prefixes in the order units-tens-hundreds. The last vowel is replaced with "-illion" at the end of a zillion word. To see how this works, let's look at the 321st zillion. The prefix for one is "un-," the prefix for twenty is "viginti-," and the prefix for three hundred is "trecenti-," so we get "unvigintitrecentillion."

As mentioned before, some of these prefixes can vary depending on where they are combined, specifically the units prefixes marked with asterisks. Referring to the above chart, they vary as follows:

1) tre- becomes tres- if the following word contains "s" or "x" in brackets
2) se- becomes ses- if the following word contains "s", and sex- if it contains "x"
3) septe- and nove- become septem- and novem- if the following word contains "m", and septen- and noven- if it contains "n"

So as a further example, the 207th zillion would be "septenducentillion." Note the inserted "n" according to the above rules.

The system also provides a mechanism to create zillions higher than the 999th. Basically, it treats a zillion number in blocks of three digits, assigns the name to each of these blocks, and combines the forms together with "-illi-." As a quick example, let's look at the 900,571,835th zillion. The 900th zillion is simply "nongentilli-," the 571st is "unseptuagintaquingentilli-," and the 835th is "quinquatrigintaoctingentillion," so we have "nongentilliunseptuagintaquingentilliquinquatrigintaoctingentillion" (!). If there is a block of three zeroes, it is written as "nilli-," so, for example, the 1,000,000th zillion is "millinillinillion."

This is a simplistic and logical way to create higher zillions, but it is by no means perfect. There is no sense of "place value" in the larger zillion words; for example, it is not immediately obvious how large each of the following zillions are, without specifically counting the "-illi-" infixes: "millinillinillinillimillimillinillion" vs. "millinillinillinillinillimillion." This is perhaps a minor problem given the size of the numbers involved!

We have by far our most complete system, simpler than the Ondrejka/Henkle system and infinitely extendable. But does it meet our criteria for a naming

system? Unfortunately, for all of its merits, there are a few areas where it does fall short. Let's again take each of our principles in turn:

1) The system must be backwards-compatible with defined English zillions
    a. PARTIALLY MET – a careful inspection of the above table will show that the prefixes combined with decillion do not result in the dictionary forms for the teen zillions. Specifically, we end up with "quinquadecillion" instead of quindecillion, "sedecillion" instead of sexdecillion, and "novendecillion" instead of novemdecillion. "Sedecillion" and "novendecillion" are arguably improvements to the dictionary forms, as *sedecim* is the standard Latin number for sixteen, and "novendecillion" is consistent with "septendecillion," but "quinquadecillion" does not harmonize with Latin *quindecim*, fifteen. This is how the system is *intended* to work, however. Conway's system disregards the Latin teen forms, and instead purely combines the Latin cardinal units (*unus, duo, tres, quattuor, quinque, sex, septem, octo, novem*) to each tens/hundreds root, with sound changes. *Quindecim* is not the inspiration for "quinquadecillion," but rather *quinque* + *decim*. So if one is willing to ignore the little-used dictionary forms for the zillions in the teens, this may not be as big of an issue as first appears.
2) The system should be logical and self-consistent, with few one-off exceptions
    a. PARTIALLY MET – the system is logical, but not self-consistent with the spelling changes. Rather than being purely based on Latin rules for sound change, they are in many cases just invented to prevent two zillions having the same form. As an example, let's look at centillion. Centillion starts with the [s] sound, even though it happens to be spelled with a "c." But prefixes combine differently with centillion than they do with other roots spelled with an "s." This is ostensibly to prevent some zillions from

having the same form; the 103rd zillion would be "trecentillion" just like the 300th zillion if this inconsistency were not in place. Instead it is "trescentillion," spelled slightly differently, but with the same pronunciation. This can be treated as a one-off exception, but is not really desirable.

3) There can be no gaps in the named numbers. If X has a name, all numbers less than X will also have names.
    a. MET
4) Words must not be excessively long, or contain awkward letter combinations; basically these should feel like they could be plausible English words
    a. NOT MET – even ignoring the combined forms for larger zillions above 999, this system does generate some very long forms, the longest being "quattuorquinquagintaquadringentillion" at 37 letters. This would make this one of the longest words in the English language, easily beating champions like "antidisestablishmentarianism" "floccinaucinihilipilification."
5) As few new words should be defined as possible, and when a new word is created, it must be etymologically sound and strongly justified
    a. NOT MET – again ignoring the zillions above 999, the system generates close to 1000 zillion names, albeit mechanically and logically. But these forms are not strongly etymologically justified, as the underlying Latin numbers do not take the form units + tens + hundreds as assumed in this system. It's debatable that the pattern of undecillion, duodecillion, etc., can be extrapolated without this happening in the underlying Latin.
6) The system should be (relatively) simple and natural, without very complex rules

a. PARTIALLY MET – the system does require the memorization of the prefixes and a few spelling rules, but it is not overly onerous.

Interestingly, a simplified version of the Conway/Guy/Wechsler system seems to be the most commonly used system on the Internet and in several semi-authoritative sources, such as Wolfram Alpha (up to "duotrigintillion") [13], and the Nasdaq Dictionary of Finance (up to centillion) [14]. In short, when the zillions below centillion are used, it is common to see them exactly mimicking the prefixes from undecillion, duodecillion, etc., without any spelling changes. This then is fully compatible with principle 1, backwards-compatibility, principle 2, no one-off exceptions, and principle 6, simplicity, but it still falls down on principles 4, word length, and especially principle 5, etymological justification. It also cannot go beyond centillion before the names are no longer unique (trecentillion being the primary example), but it is an attractive option for closing the gap between vigintillion and centillion at least.

So the Conway/Guy/Wechsler system is an innovative and authoritative system, but if we keep to the principles we set out, it is also not ideal for our purposes. Is it at all possible to define a canonical system that meets *all* of the principles? This is what I will attempt to do in the next section.

Part VI: The "Natural" System of Large Number Names

Finally, it's come time to set out a new system of number names. To start with, the system uses all of the dictionary zillion names and the Standard Latin Zillions. So we have zillions for one to twenty and all of the round numbers up to one thousand (so, thirty-seven in total), and again we need to fill in the gaps by inventing new zillion words…or do we?

The hallmark of the "natural" system of number names is that it requires only one more zillion name (to be explained later), a few general principles, and it can name every integer in a simple and natural way. If you can remember or refer to the thirty-seven Latin zillions we've already defined (a necessary evil

to retain consistency with the current dictionary system), and you can name small English numbers in the hundreds and thousands, then that is the limit of the complexity of the system.

So if the system is not inventing new zillion names in between the tens and hundreds, how can we fill in these gaps? Well, do we have unique English number names for twenty-four and one hundred sixty-eight? No – they are strung together with smaller units. If we have zillions units, tens, and hundreds defined , why would we need to create single-word names? As an example, let's take the twenty-fourth zillion. Ondrejka/Henkle would define this as "quarto-vigintillion," and in Conway/Guy/Wechsler, it is "quattuordecillion." In this new system, the zillion would simply be "vigintillion-quadrillion." A hyphen is added for aesthetics, mirroring the English number "twenty-four," but it is not required; it just improves readability.

This "vigintillion-quadrillion" format has several advantages. While the zillion name now requires two words (for such a large number, two words is actually quite reasonable!), it does not require the invention of any new words, and it happens also to better reflect the Latin *viginti quattuor*, although this is more coincidence than design; the real inspiration is mirroring smaller English numbers.

This principle can be extended to all zillions up to the thousandth zillion. A few more examples:

Index	English Number	Zillion
39	thirty-nine	trigintillion-nonillion
52	fifty-two	quinquagintillion-billion
103	one hundred three	centillion trillion
212	two hundred twelve	ducentillion duodecillion
828	eight hundred twenty-eight	octingentillion vigintillion-octillion

Note that the punctuation used mimics the English numbers, again just for clarity, not as a required element of the system.

If you recall, the zillions from two to nine are generally based on Latin number prefixes. (We do not get hung up on whether they are cardinal or ordinal or multiplicative forms like "bi-," as they have become common prefixes in English.) These zillions are good stand-ins for two to nine in our zillion names, as demonstrated above. The exception, however, is million itself. The etymology of million is based on Latin *mille* rather than any root based on "one." So it turns out million is not a good stand in for one. Instead, we must invent one more zillion name, in fact for the last Latin number that does not yet have a corresponding zillion:

Index	Prefix	Zillion
1	uni-	unillion

We justify unillion on the basis that it exactly follows the pattern established by billion with "bi-" and trillion with "tri-." It is important to stress that unillion is not used in place of million for 10^6. This would violate our backwards compatibility for such a common number name. It is required, however, in larger compound zillions that require us to represent the value one. Some examples:

Index	English Number	Zillion
21	twenty-one	vigintillion-unillion
91	ninety-one	nonagintillion-unillion
103	one hundred one	centillion unillion
531	five hundred thirty-one	quingentillion trigintillion-unillion

We can now name any zillion less than the thousandth with up to three words. How can we extend the system?

In short, we just continue the same principle, mimicking the smaller English numbers and the way that they use place value. Let's look at the example of the 2,005th zillion. The English name for the number is two thousand, five. Let's convert this to a zillion name: billion millillion, quintillion. Here the

"billion" acts as a multiplier for "millillion," representing thousand, the same way the "two" multiplies the "thousand" in the small English number.

For zillions in the *one* thousand range, it's important to start with "unillion millillion"; millillion cannot stand alone in a zillion word, the way "thousand" cannot stand alone in an English number. (This is different to the hundreds, where we have a unique name for each multiple; centillion, ducentillion, etc.) Some brief examples:

Index	English Number	Zillion
1,000	one thousand	unillion millillion
1,001	one thousand, one	unillion millillion, unillion
11,592	eleven thousand, five hundred ninety-two	undecillion millillion, quingentillion nonagintillion-billion
991,999	nine hundred ninety-one thousand, nine hundred ninety-nine	nongentillion nonagintillion-unillion millillion, nongentillion nonagintillion-nonillion

This small continuation of the pattern established for the lower zillions now allows us to name every zillion up to the 999,999th.

How do we name the millionth zillion? The place-value principle, mimicking smaller numbers, will continue to work, as long as we are able to translate numbers like "million" and "billion" into their corresponding zillions. One more simple rule and we can name all of the integers. If we need a zillion name for a *zillion*, just add another "-illion." If the zillion has multiple parts, just add "-illion" to every part. To get higher and higher numbers (to translate a "zillionillion"), just add more "-illions" as necessary. The justification here is that the format of a zillion is "Latinate number" + "-illion." The existing zillions are already based on Latin roots, so there is nothing to translate before adding the additional "-illion."

One example, the 43,912,100,121st zillion, can be created just by translating each part:

> quadragintillion-trillion billionillion, nongentillion duodecillion millionillion, centillion millillion, centillion vigintillion-unillion

The natural word order of English numbers is maintained, there is a general one-to-one correspondence between small English numbers and zillion words, and as long as we know the corresponding zillion for each English number word, we can easily create any zillion name.

It's now come time to examine this new system against the principles we established:

1) The system must be backwards-compatible with defined English zillions
    a. MET
2) The system should be logical and self-consistent, with few one-off exceptions
    a. MET – the only "exception" is the need to define "unillion" in place of "million" for use in larger zillion names, a minor, but logical concession
3) There can be no gaps in the named numbers. If X has a name, all numbers less than X will also have names.
    a. MET
4) Words must not be excessively long, or contain awkward letter combinations; basically these should feel like they could be plausible English words
    a. MET – there are no overly long words (other than when multiple "-illion" suffixes are added to astronomical-sized zillions). In fact, the longest standalone zillion word remains "quattuordecillion," a dictionary-defined word
5) As few new words should be defined as possible, and when a new word is created, it must be etymologically sound and strongly justified
    a. MET – we only invent zillion words for each single-word Latin number, and these are all based on very clear patterns
6) The system should be (relatively) simple and natural, without very complex rules

a. MET – the zillions elegantly parallel the principles established for smaller English numbers, giving them a natural and simplistic feel

With this new "natural" system of number names, the reader should be able to easily name any large number, without feeling like they are using an artificially contrived system built on shaky etymology. According to our principles, the goal of creating a suitable "canonical" system has been met.

And these number names may just prove useful some day soon.

(By the way, did you hear that an artificial intelligence has recently been able to master the over one *vigintillion-trillion* positions that are possible in the game of Go? [15] Or that a new largest prime number was discovered in January 2016, just over *three hundred septillion millionillion, quadringentillion quadragintillion-sextillion millillion, ducentillion quadrillion*? [16] I could go on…)

One final note: I would like to call attention to Robert Munafo's excellent "Large Numbers" website at http://mrob.com/pub/math/largenum.html, which was extremely useful in compiling this research and inspiring its pursuit. I highly recommend that anyone interested in this topic give it a visit.

Appendix – The Natural System of Large Number Names

1) To name large numbers in English, we need to name the sequence of "-illion" words, or zillions.
2) The first zillion is "million"; its index is one. This is followed by "billion," with the index two.
3) All subsequent zillions are named by translating the index number. The translations are as follows:

   Units: 1) unillion, 2) billion, 3) trillion, 4) quadrillion, 5) quintillion, 6) sextillion, 7) septillion, 8) octillion, 9) nonillion
   Tens: 10) decillion, 20) vigintillion, 30) trigintillion, 40) quadragintillion, 50) quinquagintillion, 60) sexagintillion, 70) septuagintillion, 80) octogintillion, 90) nonagintillion
   Teens: 11) undecillion, 12) duodecillion, 13) tredecillion, 14) quattuordecillion, 15) quindecillion, 16) sexdecillion, 17) septendecillion, 18) octodecillion, 19) novemdecillion
   Hundreds: 100) centillion, 200) ducentillion, 300) trecentillion, 400) quadringentillion, 500) quingentillion, 600) sescentillion, 700) septingentillion, 800) octingentillion, 900) nongentillion
   Thousands: 1000) millillion* (requires a preceding multiplier)

4) To convert an English number to the zillion name, substitute the above Latin-based names for English numbers word-for-word, retaining punctuation. Some subtleties:
   a. Multiples of one hundred have one name, so "two hundred" is not "billion ducentillion," just "ducentillion." This is different for thousands, where the multiplier is important: "one thousand" is "unillion millillion."
   b. "Unillion" is used for "one" in all contexts except the very first zillion, "million"
5) When naming very large zillions, it will be necessary to translate zillion words to corresponding zillions. To do so, just add an additional "-illion" to all parts. The one millionth zillion is "unillion millionillion," for example.

6) To find the index of a zillion name from a power of ten, it depends on whether you are using the short or long scale. The short scale is most common in English.
    a. Using the short scale, divide the ten power by three and subtract one. The quotient is the zillion index, and the remainder tells you if you have one, ten, or one hundred of them (remainder 0, 1, and 2 respectively).
    b. Using the long scale, divide the ten power by six. The quotient is the zillion index, and the remainder tells you if you have one, ten, one hundred, one thousand, ten thousand, or one hundred thousand of them (from 0 to 5, respectively).
7) To find the power of ten from a zillion name:
    a. In the short scale, multiply the index of the zillion by three and add three.
    b. In the long scale, multiply the index of the zillion by six.
8) Optional rule: When dealing with zillion names that require multipliers according to the normal rules, like "one unillion millillion" (the one thousandth zillion) or "one unillion unillionillion millillionillion" (the one unillion millillionth zillion), the multipliers can be removed when they meet two conditions:
    a. The multiplier has a value of one (is in the list unillion, unillionillion, etc.)
    b. The multiplier has a lower-tiered multiplier or a non-zillion number to its immediate left. For example, in "one unillion unillionillion millillionillion", both "unillion" and "unillionillion" can be removed to get "one millillionillion", but in "one unillion billionillion, unillion millionillion", only the first "unillion" can be removed to get "one billionillion, unillion millionillion." There is no loss of information here and the name is cleaner.

Citations

[1] Borgmann, Dmitri A. (1968). Naming the Numbers. *Word Ways*, *1*(1), 28-31.
[2] Ondrejka, Rudolph. (1968). Renaming the Numbers. *Word Ways*, *1*(2), 89-93.
[3] Candelaria, John. (1975). Extending the Number Names. *Word Ways*, *8*(3) , 141-142.
[4] Candelaria, John. (1976). Renaming the Extended Numbers. *Word Ways*, *9*(1), 39.
[5] Candelaria, John. (1983). A New Number Nomenclature. *Word Ways*, *16*(2), 125-127.
[6] Kongregate, Inc. (2016). AdVenture Capitalist! (version 4.4.2) [Mobile application software]. Retrieved from http://itunes.apple.com
[7] Menegus, Bryan (2016, April 18). "Guy Breaks the 5x5 Rubik's Cube Record Without Breaking a Sweat," Gizmodo Sploid. Retrieved from: http://sploid.gizmodo.com/guy-breaks-the-5x5-rubiks-cube-record-and-no-one-seems-1771562188
[8] Million. (2016). American Heritage English Dictionary (Version 8.2.7) [Mobile application software]. Retrieved from http://itunes.apple.com
[9] Conway, John H., & Guy, Richard K. (1996). The Book of Numbers. New York, NY: Springer-Verlag New York, Inc.
[10] In the new system explained here, in the short-scale, googol would be "ten trigintillion-billion" and googolplex would be "ten trillion trigintillionillion-billionillion, trecentillion trigintillion-trillion trigintillionillion-unillionillion, trecentillion trigintillion-trillion trigintillionillion, trecentillion trigintillion-trillion vigintillionillion-nonillionillion, trecentillion trigintillion-trillion vigintillionillion-octillionillion, trecentillion trigintillion-trillion vigintillionillion-septillionillion, trecentillion trigintillion-trillion vigintillionillion-sextillionillion, trecentillion trigintillion-trillion vigintillionillion-quintillionillion, trecentillion trigintillion-trillion vigintillionillion-quadrillionillion, trecentillion trigintillion-trillion vigintillionillion-trillionillion, trecentillion trigintillion-trillion vigintillionillion-billionillion, trecentillion trigintillion-trillion vigintillionillion-unillionillion, trecentillion trigintillion-trillion vigintillionillion, trecentillion trigintillion-trillion novemdecillionillion, trecentillion trigintillion-trillion octodecillionillion, trecentillion trigintillion-trillion septendecillionillion, trecentillion trigintillion-trillion sexdecillionillion, trecentillion trigintillion-trillion quindecillionillion, trecentillion trigintillion-trillion quattuordecillionillion, trecentillion trigintillion-trillion tredecillionillion, trecentillion trigintillion-trillion duodecillionillion, trecentillion trigintillion-trillion undecillionillion, trecentillion trigintillion-trillion decillionillion, trecentillion trigintillion-trillion nonillionillion, trecentillion trigintillion-trillion octillionillion, trecentillion trigintillion-trillion septillionillion, trecentillion trigintillion-trillion sextillionillion, trecentillion

trigintillion-trillion quintillionillion, trecentillion trigintillion-trillion quadrillionillion, trecentillion trigintillion-trillion trillionillion, trecentillion trigintillion-trillion billionillion, trecentillion trigintillion-trillion millionillion, trecentillion trigintillion-trillion millillion, trecentillion trigintillion-billion" !!
[11] Knuth, D. (1981). Supernatural Numbers. In: D. Klarner, ed., The Mathematical Gardner, 1st ed. Belmont, CA: Wadsworth International, pp.310-325.
[12] Rowlett, Russ (2001). Names for Large Numbers. In Rowlett, Russ, How Many? A Dictionary of Units of Measurement. Retrieved from: https://www.unc.edu/~rowlett/units/large.html
[13] Wolfram Group LLC. (2016). WolframAlpha (version 1.7.3) [Mobile application software]. Retrieved from http://itunes.apple.com
[14] Harvey, C. (2011). Glossary of Stock Market Terms. [online] NASDAQ.com. Available at: http://www.nasdaq.com/investing/glossary/ [Accessed 22 Jan. 2017].
[15] Merrett, Rebecca (2016, January 28). "AlphaGo – Google DeepMind's AI that beat Go champion for the first time," Techworld Australia. Retrieved from: http://www.techworld.com.au/article/592961/alphago-google-deepmind-ai-beat-go-champion-first-time/
[16] Mersenne Research, Inc. (2016, January 7). "GIMPS Project Discovers Largest Known Prime Number: $2^{74,207,281}-1$," Great Internet Mersenne Prime Search. Retrieved from: http://www.mersenne.org/primes/?press=M74207281